VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Odhad velikosti výsledku vícerozměrného rozsahového dotazu

Methods for Result Size Estimation of Multidimensional Range Query

2016 Robert Zaoral

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Robert Zaoral**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Odhad velikosti výsledku vícerozměrného rozsahového dotazu
Methods for Result Size Estimation of Multidimensional Range Query

Jazyk vypracování: čeština

Zásady pro vypracování:

Vícerozměrné datové struktury se využívají stále častěji pro fyzické uložení dat v databázových systémech. Tyto datové struktury umožňují dotazování na hodnot více atributů. Jejich využití znesnadňuje komplikovanější odhad velikosti výsledku rozsahového dotazu, který by mohl být využit optimalizátorem databázového systému při generování plánů vykonávání dotazu. Cílem této práce je porovnání existujících metod pro odhad velikosti výsledku rozsahového dotazu.

1. Nastudujte oblast vícerozměrných datových struktur.
2. Nastudujte metody pro odhad velikost výsledku vícerozměrného dotazu.
3. Naimplementujte zvolené metody.
4. Proveďte experimenty, metody porovnejte a vyhodnoťte výsledky.

Seznam doporučené odborné literatury:

Ju-Hong Lee, Deok-Hwan Kim, and Chin-Wan Chung. 1999. Multi-dimensional selectivity estimation using compressed histogram information. In Proceedings of the 1999 ACM SIGMOD international conference on Management of data (SIGMOD '99). ACM, New York, NY, USA, 205-214. DOI=10.1145/304182.304200 http://doi.acm.org/10.1145/304182.304200

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Michal Krátký, Ph.D.**

Datum zadání: 01.09.2013
Datum odevzdání: 29.04.2016

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 29. dubna 2016

Zde bych rád poděkoval především vedoucímu práce doc. Ing. Michalu Krátkému, Ph.D. za pomoc, trpělivost, věcné připomínky a vstřícnost při konzultacích. Mé poděkování patří též Bc. Pavlíně Zárubové za tlumočení do znakového jazyka a Mgr. Jitce Zemanové za jazykovou revizi práce. Bez nich by tato práce nevznikla.

Abstrakt

V moderních databázových systémech se využívají nejen vícerozměrné dotazy, ale i klasicky jednorozměrné dotazy. S těmito vícerozměrnými dotazy se v provozu databáze musí vyrovnat. Pro efektivní zpracování těchto dotazů plán na optimalizaci dotazů spoléhá na vícerozměrné selektivity odhadu techniky. Tato technika zase typicky spoléhá na histogramy. Základním stavebním kamenem histogramu je detekce regionů s vyšší hustotou než jejich okolí. U výběru vzorků je získání data populace na dotazování respondentů, v kvantitativním výzkumu šetření, v statistice atd. Nalezení shluků ve vícerozměrném prostoru může být vytvoření vysoce kvalitních histogramů. Ukázali jsme principy, pojmy a problémy histogramů a jejich řešení pomocí různých metod. Cílem této diplomové práce je vytvořit rozsáhlejší řešení pro odhad velikosti výsledku, jak jednorozměrná selektivita, tak vícerozměrný histogram. Pro řešení výsledků a jejich porovnání mezi skutečnosti a odhadem jsou aplikovány v implementačním prostředí programovacím jazykem.

Klíčová slova: selektivita, odhad, histogram, optimalizace dotazu, visual studio, jedno rozměrné, více rozměrné prostory

Abstract

In modern database systems are used multidimensional queries, but also a classic one-dimensional queries. With these multidimensional queries in the database operation to cope with. For efficient processing of these queries, query optimization plan relies on multidimensional selectivity estimation techniques. This technique typically relies on histograms. The basic building block of the histogram is to detect regions of higher density than their surroundings. For the sampling is to obtain data on the population poll respondents in quantitative survey research, statistics and so on. Finding clusters in a multidimensional space may be creating high-quality histograms. We showed principles, concepts and histograms problems and their solutions using different methods. The aim of this thesis is to create a larger solution to estimate the size of the result as one-dimensional selectivity and multidimensional histogram. To tackle the results of a comparison between facts and estimates are applied in the implementation environment programming language.

Key Words: selectivity, estimate, histogram, query optimization, visual studio, one dimension, multiple-dimension

Obsah

Seznam použitých zkratek a symbolů

$\mathbb{R}$	–	množina reálných čísel
IS	–	Informační systém
SVD	–	Singular value decomposition
DBMS	–	Database Management System
SŘBD	–	Systém řízení báze dat
MaxDiff	–	Maximum difference scaling
GENHIST	–	Generalized histograms
MLGF	–	Multi Level Grid File
API	–	Application Programming Interface
GUI	–	Graphical User Interface
WPF	–	Windows Presentation Foundation

Seznam obrázků

Seznam tabulek

1 Úvod

Toto téma jsem si vybral volbou z rozsáhlého seznamu zadání v informačním systému EDISON ve VŠB-TU Ostrava. Toto téma bylo v IS volné, to byl ten první důvod a druhý důvod vycházel z toho, že toto téma je odbornou kapitolou v oblasti informatiky, kterou celou dobu nejenom studuji, ale učím se nové věci. Sbírám také nové zkušenosti ve firmě, ve které pracuji. Někdy pracuji také dotazech pomocí syntaxe SQL v databázovém prostředí. Z pohledu tohoto tématu mám pocit, že budu probírat zejména odbornou i praktickou látku z kategorie statistiky, v níž se ukáže základní pohled na histogramy a analýzu získaných dat z reálného světa.

Dále se budu snažit věnovat probírané teoretické oblasti ze statistiky a její srozumitelné struktuře a jejímu rozsáhlému obsahu a pak se budu dále věnovat praktickému prostředí, zejména implementační atmosféře v programovacím jazyku. V oblast implementace 7 se provádí sběr dat na vstupní prostředí, zpracovávají se data na základě výpočtů a do výstupu se vezme konečný výsledek pro odhad a vykonávaní dotazů na prostředí uživatelského rozhraní. Nejdříve budu popisovat podrobnost odhadů, jak se to dělá, proč využíváme tyto odhady, co nakonec dokazují pro konečný výsledek po experimentu 7.4. Tento výpočet odhadů můžeme použít podle různých popsaných všeobecných metod a dostupných, které jsem zařadil do kapitoly 3. Metod pro výpočet odhadu v informatice existuje mnoho. Ve své práci budu popisovat základní zvolené postupy pro teorii a praxi na implementaci a to „výběr vzorků“ pro kategorii jednorozměrnou selektivitu 4.3 a „STHoles“ pro vícerozměrnou selektivitu 5.4.

Dále u každé vybrané metody je zpracován popis a princip funkcionality a výpočtu podle vzorce z dostupné zahraniční literatury. Vytvořím a provedu proces u implementace do podsekce experimentu. V zóně implementace pro vzorkování se bude implementovat aplikace. A pro STholes se budu pokoušet získat zdroje literatury a knihovnu do implementace ze třetí strany (například MSDN z webové stránky Microsoftu) a podle toho budu částečně implementovat tak, aby výsledek byl pro přehled dostatečný. Vytvoření vlastní techniky STholes používám pro dvě dimenze v rámci vícerozměrného rozsahového dotazu.

U implementace jsem zvolil nástroje vývojové aplikace Visual Studio s programovacím jazykem C#. Dále jsou popisy, jak probíhá proces práce na datové sadě a sekvence rozsahových dotazů od vstupu až po výstup 7.4.3.

V rámci úvodu bych chtěl také uvést, že jsem neslyšící student, je pro mě obtížné psát bezchybně česky. Čeština je pro mě cizí jazyk, proto je pro mě problematické, vyjadřovat se gramaticky správně, zejména v odborných textech. Jak jsem již psal v mé bakalářské práci o problematice informatiky pro neslyšící, český jazyk není mým mateřským jazykem, proto jsem se snažil nechat svou práci opravit, aby byly v souladu se správnou gramatikou českého jazyka.

2 Historie

S vývojem odhadu u statistiky standardizované velikosti výsledku začal v 20.století Jacob Cohen. Další významnou osobností, která se zajímala o totéž a upřesnila metodu odhadu v oblasti statistiky, je Larry V. Hedges.

2.1 Jacob Cohen (1923 – 1998)

Jacob Cohen byl statistik a psycholog, který žil ve Spojených státech amerických. Zajímal se nejvíce statistiku. Nakonec pomohl položit základy pro aktuální statistické meta-analýzy a metody pro odhad. Vynalezl vlastní metody, které se nazývají se Cohen's kappa a Cohen's d.

Získal magisterský titul v roce 1948 a doktorát na psychologické klinice univerzity v New Yorku v roce 1950. Na této univerzitě pracoval jako instruktor a byl povýšen na profesora o 10 let později, když pracoval jako koordinátor univerzity pro kvantitativní psychologii. Dr. Cohen strávil ve výzkumu na univerzitě v New Yorku 44 let.

Věnoval se psání článků a knih o statistické analýze a řešení různých nedostatků v metodách, které čerpal ze svého psychologického výzkumu.

Jacob Cohen vymyslel nové statistické metody s více proměnnými v době, kdy do svého času investoval jeho výpočet. Jedna statistika byla založena na jeho výzkumu psychiatrického vyšetření, které známe jako Cohenova kappa, bylo široce přizpůsoben pro bio statistiku a lékařské výzkumy.

Po jeho formálním odchodu z psychologické katedry na univerzitě v roce 1993, s věnoval výzkum a pracoval jako statistický poradce v centru pro klinické a behaviorální studium.

Dr. Cohen svou strategii nasadil adekvátní velikosti vzorkování do učebnici v nakladatelství Erlbaum [30]. Jeho knihy v oblasti statistiky a metody odhadu byly mezi nejprodávanějšími.

Byl zvolen kolegy do mnoha profesních organizací, včetně americké asociace pro rozvoj vědy, americké asociace psychologie a americké statistické společnosti. Pracoval také jako prezident společnosti pro multivariantní experimentální psychologii.

2.2 Larry V. Hedges

Larry Vernon Hedges je výzkumník v statistických metod pro meta-analýzu a vyhodnocení politiky vzdělávání. Je profesorem statistiky, vzdělávání a sociální politiky na univerzitě Northwestern.

Larry v. Hedges se připojil k fakultě na Northwestern v roce 2005 a je vedoucí v oblasti vzdělávacích statistik a vyhodnocování. On je jedním z osmi v správní radě profesorů na univerzitě Northwestern v nejvýznamnější akademické pozici. Je držitelem ocenění v statistice, psychologii a sociální politice. Dříve působil jako profesor na univerzitě v americkém městě Chicagu.

Pracoval na vývoji statistických metod pro meta-analýzu (statistickou analýzu výsledků několika studií, které kombinuje svá zjištění) v sociálních, lékařských a biologických vědách.

(a) Jacob Cohen

(b) Larry Venom Hedges

Obrázek 1: Jacob Cohen a Larry Venom Hedges

Je autorem či spoluautorem řady článků a knih o statistických metodách, která používá kombinaci statistických výsledků z různých studií. Navrhl také několik odhadů velikosti výsledku a jejich vlastnosti. Úspěšně vedl také mnoho výzkumů různých zdrojů do školy. Publikoval i několik prací např. [32] [33].

Jeho výzkumné zájmy jsou ve vývoji a aplikaci statistických metod pro sociální, zdravotnické a biologické vědy. Další významné oblasti výzkumu jsou návrh a analýza sociálních experimentů. Jeho práce na rozvoji statistických metod v obou těchto oblastech a jejich využití k výzkumu politiky, zejména vzdělávací politiky.

3 Metoda odhadu velikosti výsledku dotazu

Když pracovník potřebuje zjistit svou odpověď na dotaz s tímto příkladem, kolik kusovníků je ve výrobě za celou pracovní dobu v rozmezí 12 měsíců v průmyslové továrně, zadá do databáze požadovaný dotaz, ale byl informován, že záznam v této databázi obsahuje miliony položek, a proto je složité získat správnou odpověď v krátkému času. Abychom získali průměrný výsledek v co nejkratší době využijeme možnost metody pro odhad.

Tato metoda pro odhad, která vygeneruje velikost výsledku z databáze po zadání dotazu, ukáže odpovídající výsledek jako vyhodnocení dotazů. To závisí na objemu datové sady, což časově ovlivní výpočet statistických metod. Toto zadání na dotaz používá uživatel, který potřebuje znát odpověď ze statistického důvodu i výsledek obsahující záznamy nebo výsledky v (barevných) histogramech v databázi a jeho odpověď může být časově neefektivní a rozsáhlá kvůli velikosti či složitosti selektivity. Důležitý faktor pro výpočet selektivity je plán. Efektivní plán výrazně ovlivňuje snížení nákladu a vyšší přesnost odhadu selektivity. Pro odhad selektivity se obvykle pohybuje distribuce dat z teorie statistiky. U dimensionality existují dvě třídy tohoto problému.

Třída dimensionality má jednorozměrnou selektivitu odhadu a vícerozměrnou selektivitu odhadu. Multidimenzionální selektivita je poměrně náročnější než jednorozměrná selektivita. Práce s multidimenzionální selektivitou vyžaduje několik aplikačních nástrojů. Proto se snažím pátrat po těch autorech, kteří tento problém již vyřešili pomocí statistických metod a technik. Díky jejich způsobu se dostane efektivní a přesný výpočet do histogramu(odhadu) a jeho menší chybovost.

Pro optimalizaci dotazu na dolování dat a datové sklady se nabízí technika hodnocení **top-k**. Následující ukázkové příklady týkající se dotazu top-k se popisují v článku [1], na který odkazujeme. Tento model dotazu v tradičním relačním systému, odpověď výběru dotazu je množina, v které leží n-tice prvků. V kontrastu odpověď na dotaz top-k je uspořádaná množina n-tic prvků, kde toto uspořádaní úzce reflektuje každý z n-tice prvků, odpovídající danému dotazu. Tato sekce definuje náš model dotazu přesně.

Vezměme si relaci R s atributy $A_1, ..., A_n$. Top-k dotazů nad R jednoduše specifikuje cílové hodnoty pro atributy v R. Tak dotaz je přiřazení hodnot $v_1, ..., v_n$ na atributech $A_1, ..., A_n$ v relaci R. V tomto článku se zaměříme na top-k dotazů na spojitých atributech (například věk, mzda). Předpokládáme, že hodnoty těchto atributů jsou normalizovány reálná čísla mezi 0 a 1.

Opět mějme relaci $R = (A_1, ..., A_n)$. $A_1, ..., A_n$ jsou to atributy reálných čísel v rozmezí 0 až 1. Poté daný dotaz $q = (q_1, ...q_n)$ a n-tice prvků $t = (t_1, ...t_n)$ z relace R. My budeme definovat funkcionalitu pro dotaz q v n-tice množiny t použití některé z následujících dvou agregačních funkcí $Min(q, t)$, $Sum(q, t)$ a funkce $Euclidean(q, t)$ [1].

Mapování strategií a hodnocení výkonnosti dotazu prostřednictví top-k [2]. Model dotazu pro mapování strategie má skoro podobný princip jako předchozí citace, tedy zejména tři distanční

funkce. U mapování existuje statické i dynamické řešení. V oblasti statické se vyhodnocení strategie zpracování dotazu skládá z následujících tří kroků.

- **Search** - Dotaz top-k q nad relací R používáme multidimenzionální histogram H k odhadu hledání vzdálenosti dq. Očekává se, že region $reg(q, dq)$ obsahuje všechny možné n-tice prvků ve vzdálenosti dq nebo nižší od q zahrnující n-tice prvků.
- **Retrieve** - Načítáme všechny n-tice prvků v regionu $reg(q, dq)$ pomocí rozsahu na dotazu, který uzavírá tuto oblast co nejtěsněji.
- **Verify/Restart** - Pokud existují přinejmenším n-tice v regionu $reg(q, dq)$, tak vrátí k n-tice s nejnižší vzdáleností. Jinak se vybere nejvyšší hodnota distance a restartuje se procedura.

Technologie dynamického mapování strategie se představí v parametrickém mapování, které odvodí jednoduchý postup zvoleného parametru, který vede k nejlepší strategii pro dané vytížení dotazů. Můžeme provést stejnou praxi jako v předchozí popsané technice. Stejné datové sady fungují nejlépe pro dotaz q, někdy závisí na specifikaci dotazu q. Vzhledem k tomu, že výsledná strategie mapování bude záviset na konkrétním zatížení (oproti statické technice) dynamicky. Detailní popis v rámci tohoto výkladu je uveden v článku citace [2].

Tree estimation [3] se představí jako algoritmus pro efektivní a účinné odhadování přibližování nevybraných uzlů v procesu vyhledávání. V tomto přístupu je uzel, který navštívíme jeden po druhém a odhadujeme jeho blízkost. V případě, že odhadovaná blízkost není nižší než nejvyšší blízkost kandidátských uzlů, pak uzel je vybrán pro výpočet přesné blízkosti. V opačném případě bychom vynechali následně přesné přiblížení výpočtu navštívených uzlů. Tento přístup, na základě jedné šířky prvního vyhledávacího stromu, poskytuje horní odhady ohraničující skóre navštívených uzlů. V této části prezentuje druh odhadu a to je notace, formálně přiblížení odhadu, výpočetní inkrement a pak následný přístup ke kumulativnímu odhadu v procesu vyhledávání.

Díky těmto technikám top-k můžeme zjistit přesný odhad velikosti výsledku. Můžeme prezentovat tuto techniku, které říkáme nový algoritmus. Tento nový algoritmus odpovídá výsledku dotazu pomocí prostředí top-k.

Další technika pro optimalizaci dotazu, která odkazuje na více atributů v relaci tabulky, kdy velikost výsledku dotazu závisí na společných datech z distribuce dat, byla reprezentována jako **vícerozměrný prostor** [4]. Tento druh techniky budu nadále pro moji rozšířenou práci vyhledávat. Dále v databázi nebo kdekoliv na prostorovém úložišti je datová sada, která může obsahovat velké množství atributů s reálnými hodnotami. Úložiště mohou obsahovat tři druhy typu dat – prostorové, časové a multimediální, tyto objekty jsou prezentovány jako funkce vektoru na bázi [6] úložiště prostředí pro multimediální data(například rychlost větru a srážek, klimatické údaje, atd.) [10].

Ještě dále existuje možnost řešení různými metodami k danému problému odhadu selektivity. Máme metody pro jednodimenzionální selektivitu – **parametrická metoda** [7], **neparametrická metoda** [5], **metoda aproximace křivky** [8], **metoda výběru vzorků** [9]

v kapitole 4.3, atd. Tato neparametrická metoda je pro výpočet odhadu výhodnější, protože je schopna v průběhu výpočtu mít nízkou chybovost, nejen to ale je i blízko k distribuci dat. Tak jsme si už vysvětlili, co jsou metody a k čemu slouží, teď se zkusíme vypořádat zejména s odhady z vícerozměrné selektivity. Tento odhad z vícerozměrného prostoru je naproti jednodimenzionální selektivitě obtížný a složitý pro výpočet funkce, vyžaduje to speciální metody a techniky, více místa paměti v úložišti pro objem z multimediální datové sady. K vytváření vícerozměrného histogramu je dobré mít na sobě atributy nezávislé. Pokud jsou tyto atributy nezávislé, tak nejen to bude vyšší optimalizace pro dotazy ale i přesné odhady za využití statistických výpočtů a funkcí. Jak vypadá vícerozměrná selektivita, můžeme brát v úvahu, že je to součin jednorozměrných odhadů(matice). Následující jsou běžné názvy efektivní techniky pro nalezení odhadu pro vícerozměrné selektivity obsahující datové sady s reálnými hodnotami. Tato technika se jmenuje **GENHIST** [11] [12], která byla popsána po předběžném experimentálním pokusu. A ještě dalších několik technik bylo navrženo za účelem snížení chyby v odhadu. Pro vícerozměrné selektivity odhadu jsou k dispozici metody: **MLGF** [13], **SVD** [15] [16], **Hilbert numbering** [17], **PHASED**, **STHoles** [20] a **MHIST** [4].

Optimalizace dotazů je nezbytnou součástí každého systému pro správu databáze. Optimalizátor spoléhá na přesné odhady velikosti dílčích dotazů. Za tímto účelem optimalizace dotazů odhadne selektivitu dotazu predikát, tj, počet záznamů, které splňují predikátu. Predikát může označovat několik atributů. Jsou-li tyto atributy korelovány, pro statistické údaje o jejich společnou distribuci je zásadní přijít s dobrým odhadem. Odhad je hodnota, která se získá zpravidla odhadech na konkrétní vzorek. Estimátor (odhad) je pravidlo, kterými se vypočítává odhad na základě informace ve vzorku. Odhad pro samplování je třeba vzít v úvahu při vyšetření celé populace, vyžaduje podrobné pokyny a nároky na čas, nebo rozsáhlou příručku, která je provedena pro každou položku. Rozumné odhady jsou přijatelné jako alternativa k přezkoumání celé populace. Pro použití odhadu ze samplování je třeba brát ohled na tu populaci, která obsahuje zaznamenanou hodnotu správně v určitém rozsahu. Vícerozměrné histogramy obsahují v statistice více atributů.

4 Jednorozměrná selektivita

Obvykle se používají jednorozměrné histogramy. Následné techniky jsou stále široce využívány [21].

4.1 Parametrická metoda

Parametry populace jsou konstantní, parametry náhodného výběru jsou náhodné proměnné řídící se výběrovým rozdělením. Formulujeme úlohu, ve které hledáme odhady neznámých parametrů rozdělení. Pokud známe při analýze konkrétní výběrovou charakteristiku, pak jsme schopni odhadnout parametr celé populace.

Bodový odhad
Z určitého rozdělení máme určitý výběr $X_1, X_2, \ldots, X_n$. To závisí na neznámém parametru, který se označuje Θ. Odhadem parametru je pak výběrová charakteristika, která nabývá hodnot důvěrných neznámému parametru. Protože odhad je funkcí náhodných veličin, je také náhodnou veličinou. Pokud jeho střední hodnota se rovná hledanému parametru, tak je to nestranný odhad. Pro konstrukci bodových odhadů hodnot parametrů pravděpodobnostních se nejčastěji používají **metody maximální věrohodnosti** [18] nebo **metody momentů** [19].

Nejlepší bodový odhad

- střední hodnota μ je průměr $\overline{x}$
- rozptyl σ^2 je výběrový rozptyl s^2
- směrodatná odchylka σ je výběrová směrodatná odchylka s
- relativní četnost π je výběrová relativní četnost p

Metoda maximální věrohodnosti
Je založena na vlastnostech sdružené hustoty či pravděpodobnostní funkce. Tato metoda dává poměrně kvalitní výsledky a je často používána. Základním pojmem je zde tzv. věrohodností funkce. Tato funkce je založena na podmínce maximalizace, což je sdružená hustota pravděpodobnosti daného náhodného výběru. Protože má v řadě případů hustota exponenciální průběh, používáme místo věrohodností funkce její logaritmus. Maximálně věrohodný odhad je řešením soustavy věrohodností nějakých rovnic. Ovšem bráno jako funkce neznámých parametrů.

Metoda momentů
Je založena na jednoduché metodě pro porovnání momentů základního souboru a výběrové sady s odpovídajícími teoretickými momenty předpokládaného rozdělení pravděpodobnosti (například Poissonovo rozdělení, exponenciální rozdělení, normální rozdělení). Nelze jednoznačně

rozhodnout, která z metod dává lepší výsledky, obecně však nedává příliš kvalitní efekt. Rozhodování provádíme podle konkretní situace, nejčastěji rozhoduje jednoduchost získaných výběrových vzorů. Metoda vede na řešení soustavy takového počtu rovnic, kolik je neznámých parametrů. Při řešení odhadů získané metodou momentů může být ve výjimečném případě totožné s odhady získanými metodou maximální věrohodnosti.

Intervalový odhad

Odhad v základním souboru je interval $\langle t_D, t_H \rangle$, ve kterém leží skutečná hodnota hledaného parametru s předem určenou pravděpodobností (spolehlivostí), kterou označujeme $(1-\alpha)$. Věnujeme se nyní intervalovému odhadu nejdůležitějších statistických veličin, střední hodnoty a rozptylu.

Intervalový odhad **střední hodnoty** počítáme jinak, jestliže **známe nějaký rozptyl** základního souboru a jinak, když konkrétní populační **rozptyl neznáme**. V případě, že neznáme rozptyl, ale i předem **známe směrodatnou odchylku**, můžeme použít při odhadu střední hodnoty. Nebo můžeme odhalit tuto hodnotu i v případě, že máme **dostatečně velký výběr** ($n \geq 30$) a **směrodatnou odchylku neznáme**.

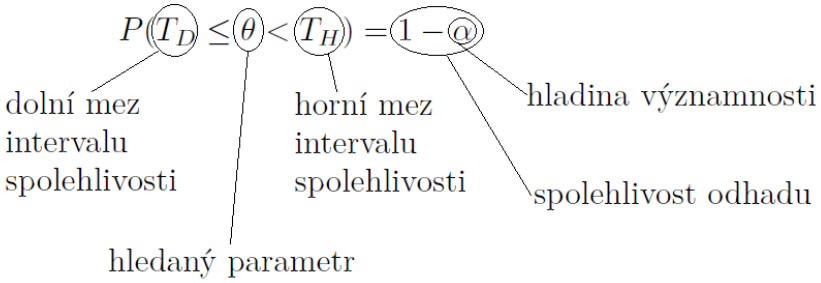
Tento odhad jsem vyvinul ve své aplikaci, která je popisována funkcionality, a je odkazována v praktické části implementace.

Pokud zjišťujeme střední hodnotu nebo rozptyl základního souboru na základě nějakého výběru z něj, konečný výsledek udáváme v intervalu, protože není možné určit přesnou hodnotu. Nemůžeme ale říct, že průměrná výška obyvatel ČR se bude na 100% nacházet v intervalu 150 až 200. Výsledkem je interval, ve kterém se bude nacházet hledaná proměnná, a i to pouze s určitou pravděpodobností. Intervalové odhady používají běžné postupy **interval spolehlivosti**, **intervalový odhad parametru**, **spolehlivost odhadu** a **hladina významnosti** [18].

- Interval spolehlivosti - pro parametr Θ se spolehlivosti $1-\alpha, kde \alpha in \langle 0, 1 \rangle$, je taková dvojice statistik, obrázek 2, $(T_D; T_H)$, že $P(T_D \leq \theta < T_H) = 1 - \alpha$.
- Intervalový odhad parametru - θ se spolehlivostí $1 - \alpha$ je interval $\langle t_D, t_H \rangle$, kde t_D, t_H jsou hodnoty statistik T_D, T_H na daném statistickém souboru $(x_1, \ldots, x_n)$. Intervalový odhad je jednou z realizací intervalu spolehlivosti.
- Spolehlivost odhadu - $(1-\alpha)$ předem určená pravděpodobnost (např. 95% pro průmyslovou složku, 99% pro biomedicínskou zónu).
- Hladina významnosti - označí se $\alpha(\alpha - testem)$, $\alpha \in (0, 1)$, rozumíme test, u kterého pravděpodobnost chyby nepřekračuje hodnotu α.

4.2 Neparametrická metoda

Neparametrické metody se nespoléhají na odhady parametrů charakterizujících proměnné rozdělení v základním souboru. Proto se tyto metody někdy označují jako metody s volnými rozděleními. Neparametrické metody pracují s četností (např. Chí-kvadrát test nezávislosti) nebo



Obrázek 2: Intervalové odhady

s pořadovými čísly, které se přidělí původními údaji (např. Kruskal-Wallisův test). Hodí se to při použití malých vzorků (např. výběrový vzor n do 30) ze základních souborů s normálním rozdělení. Oproti počtu množství vzorků je parametrická metoda. Na druhou stranu v případě velkých vzorků (např. stovky výběrů) je třeba brát ohled na to, že zřejmě používáme tyto metody, protože se počítají snadněji než neparametrická metoda. Neparametrické testy se mohou používat pro data z intervalové a poměrové škály, pokud je převedeme na kategorizaci nebo vhodné uspořádání.

Chí-kvadrát test
Test dobré shody používáme obecně k testování shody četností (především u nominálních znaků - kategoriálních dat), ale můžeme ho použít i k otestování shody rozdělení četností u znaků kvantitativních, a to metodou porovnání distribuční funkce sledované spojité náhodné veličiny s distribuční funkcí normovaného normálního rozdělení.

Kruskal-Wallisův test
Kruskalův-Wallisův test je zobecněním neparametrického Mannova-Whitneyho testu pro více než dvě srovnávané skupiny. Stejně jako Mannův-Whitneyho test tak netestuje shodu konkrétních parametrů, ale shodu výběrových distribučních funkcí srovnávaných souborů s tím, že klíčovým předpokladem je zde nezávislosti pozorovaných hodnot.

Mann-Whitney(ův) test
Je neparametrická obdoba dvouvýběrového t-testu nulové hypotézy. Oba z výběru ze vzorků pocházejí ze stejné populace oproti alternativní hypotéze, a to zejména, že konkrétní populace má větší tendenci než druhý výběr.

4.3 Výběr vzorků

Tento výběr vzorků (někdy říkáme samplování) patří do kategorie statistického šetření nebo výběrové statistické metody. Statistika je věda o sběru, zpracování a vyhodnocování dat. Datová

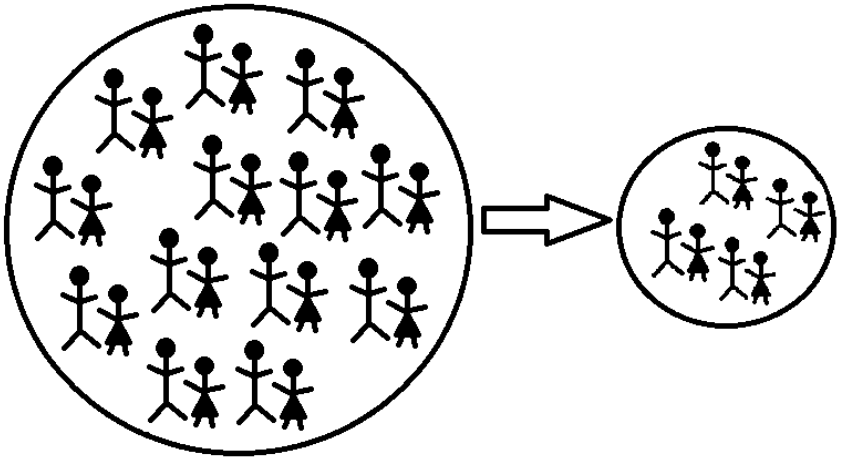
sada bude chápána jako základní soubor z celé populace. Populace je zadána přesným stanovením jejích prvků. Prvky jsou buď výčtem nebo vymezením některých společných vlastností. Bez statistického pravidla by nebyly náhodně vybrány ze záznamů ani nelze extrapolovat výsledky na celou populaci. Pracovníci ve výzkumu nebo uživateli v praxi většinou nemají čas, peníze ani energie na to, aby mohli prozkoumat všechny statisíce kolekcí z datové sady, vztahující se k problému z analýzy. Proto řešení je takové, že údaje v datové sadě jako základní soubor obsahuje všechny možné statistické jednotky, provedeme relativně výběr do výběrového souboru, který obsahuje vybrané statistické jednotky a jejich parametry pak lze zpracovat na zlomek. Pro statistické samplování je hlavní používat statistické teorie a zákony pravděpodobnosti a vyhodnotit vzorek. Záleží na tom jaké populace jsou v základním souboru, může to být třeba pohlaví osob, jiná skupina zvířat pro účely ochránců zvířat nebo různé dotazy od respondentů za účelem auditu. Každá položka v populaci má již dříve známou pravděpodobnost pro výběr. Položky jsou vybrány náhodně. Výsledky statistického vzorku lze extrapolovat na celou populaci s určitým stupněm spolehlivosti.

Parametr (populační charakteristika) je číselná charakteristika populace. Tento parametr má jistou pevnou číselnou hodnotu, ale jeho hodnota je obecně neznámá. Právě statistika umožňuje odhadovat tyto parametry pomocí výběrových statistik. Pokud se provede výběr z populace, potom se pomocí naměřených dat vypočítá výběrová statistika. Například průměrná výška desetiletých kluků v Česku, odhad průměrného obsahu tuku v polotučném mléku. K parametrům základního souboru patří například střední hodnota, rozptyl, směrodatná odchylka, atd. Jak jsem již popsal, tento parametr u základního souboru má konstantní hodnotu, ve skutečnost ji neznáme ani znát nebudeme. Možností je, že můžeme najít příslušné parametry populace ve výběrové statistice. Z tohoto pojmu vyplývá, že výběrová charakteristika je definována jako vhodná funkce pro náhodný výběr. Pokud skutečně statisticky zjišťujeme nějaký jev, někdy se stává, že objem souboru je tak velký, že pochopitelně budeme mít trochu potíže zjistit přesnost jevu. Zkusíme si představit testování všech součástí. Nemůžeme otestovat všechny, ale jen nějaký vzorek (výběrový soubor). Pro nás je zřejmě důležité to, že základní soubor má svoje statistické jednotky jako jsou rozptyl a průměrná hodnota. Stejně i tento výběrový soubor má nějaký průměr resp. rozptyl a proto chceme na základě dat určit průměr nebo rozptyl základního souboru.

Vznikne jednoznačná metoda „výběr vzorků“.

4.3.1 Šetření a postup

V praxi se většinou setkáváme se základními soubory, které mají velmi vysoký obsah. Před analýzou a zpracováním dat se rozhodneme pro určitý typ šetření a to buď vyčerpávající nebo výběrové. Proč musíme realizovat práci na jednom z typu šetření? Informace o populaci získáváme prostřednictvím statistického výzkumu. Následují texty, kde získáme množství dat z populace při výběrů v reálném čase.



Obrázek 3: Populace a vzorek

- **Vyčerpávající (úplné) šetření** - to jsou všechny jednotky v základním (statistickém) souboru včetně populací. Jedná se o nákladné práce i delší čas práce personálu. Může to být nedokončená realizace v průběhu analýzy dat. Pro praxi je to nedostatečné a nevýhodné. Patří do nepřesnosti, to znamená, že se by musela realizovat každá jednotka v statisíci záznamů populací.
- **Výběrové (neúplné) šetření** - jedná se o prošetření vybraných jednotek v statistickém souboru. Z těchto souborů pak můžeme usuzovat více či méně na vlastnosti celé populace. Naproti vyčerpávajícímu šeření je nejen méně nákladné, ale lze i v kratším čase analyzovat a realizovat očekávané výsledky. Tato šetření se používají například v podniku při ověřování procenta zmetků z výroby. Kvalitu provedení výběrového šetření můžeme získat z informací mírou objektivnosti.

Při statistickém šetření je výzkumný pracovník pouze pozorovatelem, který zasahuje co nejméně do průběhu šetření. Pokud sledujeme existující znaky u všech jednotek populace, tak provádíme vyčerpávající šetření. Mohu uvést názorný příklad - úplné šetření na demografických souborech je provádění soupisů, např. soupis pacientů v lůžkových zařízeních. Pro soupis obyvatelstva se používá sčítání lidu a v našich zemích se provádí v intervalu deset let. Tato šetření také můžeme provést při evidenci hlášených nemocí, při sledování důležitých demografických jevů, jako je v rámci soupisu obyvatelstva, narození nebo úmrtí. Vzhledem ke stanoveným cílům bývá úplné šetření na rozsáhlejších populacích organizačně, ekonomicky a časově tak náročně, že je nelze uskutečnit. Proto zpravidla přistupujeme k výběrovému šetření. Při tomto šetření zjišťujeme požadované vlastnosti pouze u některých prvků populace, které vytvářejí výběr.

4.3.2 Typ výběrových šetření

Mezi druhy výběrových šetření zařazujeme tabulku záznamů z ankety, záměrného výběru a náhodného výběru.

- Anketa je takový průzkum, který dodává otázky ve formě dotazníku s žádosti o jejích vyplnění a vrácení pro respondenty. Respondenti můžou odpovídat dle zadaných otázek hlasováním nebo svým názorem v odpovědi. Je to nesystematické šetření a získané informace z anketního šetření nelze zobecňovat, protože odpovědi (výjimka v hlasováním bodů) u jednotlivých respondentů jsou různé. Nelze definovat populaci, ke které se výsledky ankety vztahují, protože nejsou to všichni čtenáři určitých novin, časopisů, jsou to právě jen ti, kteří zodpověděli anketu.

- Záměrný výběr provádí zkušený odborník ve výzkumu, který vybere jednotky podle svého uvážení, nejen z finančních důvodu, ale vybere i pravidla pro statistické zkoumání. Je založen pouze na úsudku výzkumníka o tom, co by mělo být pozorováno a o tom, co je možné pozorovat. Někdy se tomu říká účelový, úsudkový výběr.

- Kvótní výběr usiluje o strukturální shodu výběrového souboru se základním souborem. Může být schopen dodat své chování vzorku na předpověď chování populace, musí struktura vzorku imitovat složení populace tak přesně, jak je možné např. populace obsahuje 51% žen, vybere se totéž 51% žen do vzorku, a když je v populaci 12% osob nad 65 let věku, tak se vybere stejné procento starých osob do vzorku, apod. Výběr statistických jednotek do kvótního výběru probíhá na základě kritérií daných kvótou. Takovým kritériem může být například zastoupení jednotek podle pohlaví, věku, vzdělání osob.

- Náhodný výběr je základním a nejpoužívanější typ výběru. O tom, které jednotky budou zařazeny do výběrového souboru rozhoduje náhoda a ne záměr nebo úsudek vybírajícího. Takový výběr, který má pro všechny jednotky základního souboru stejnou pravděpodobnost, že do výběru bude vybrán. Každý element musí mít stejnou šanci, že bude vybrán do vzorku. Z výběrového souboru můžeme zobecnit získané charakteristiky na základní soubor za pomoci metod matematické statistiky.

Realizace statistické analýzy se někdy řídí takovými postupy.

1. Slovní vyjádření k řešení - koho se daný problém týká, co chceme zjistit, jaké konečné výsledky chceme očekávat.

2. Sběr dat - ukládaní informací, především za pomoci statistického šetření.

3. Analýza - rozbor shromážděných dat, vedoucí k získání potřebné informace.

4. Vyhodnocení - získané a rozpracované informace.

4.3.3 Klíčový pojem

Samplování je výběr a implementace v rámci statistického oboru za účelem odhadu vlastnosti populace. Výběr vzorků je podmnožinou populace, která získá prostřednictvím produktového

procesu, případně náhodného výběru nebo výběr na základě určitého souboru dle kritéria, pro účely vyšetření vlastností základního souboru (populace).

4.3.4 Výběrová charakteristika

U výběrové charakteristiky můžeme výsledek vypočítat pomocí číselných charakteristik - střední hodnota, rozptyl, směrodatná odchylka, modus. Při náhodnému výběru jsou tyto výběrové hodnoty hodnotami náhodných veličin. To znamená, že základní soubor při vypočítání přes tuto charakteristiku jsou vlastně funkcích těchto náhodných veličin a samy o sobě jsou tedy také náhodnými veličinami. Tyto funkce budeme používat při odhadování charakteristik nebo parametrů. Je to obecně náhodná veličina. Dále uvedeme příklad, jak to funguje. Mějme základní soubor o rozsahu N, zajímá nás hlavně znak X (např. objem červeného vína v lahvi). Ze základního souboru vybereme nějaké jednotky. Výběr každé jednotky můžeme považovat za náhodný pokus. Zkoumaný znak je vlastně náhodná veličina. Tato náhodná veličina má dva typy a to buď pravděpodobnostní funkce nebo funkce hustoty pravděpodobnosti. U každé jednotky, která se dostane do výběrového souboru, zjistíme hodnotu zkoumaného znaku $x_i (i = 1, 2, \ldots, n)$.

4.3.5 Histogram

Histogram je sloupcový graf, který zachycuje intervaly na vodorovné ose. Do histogramu se data z distribuce vypočítají na základě techniky a jsou rozčleněna. Na svislé ose je počet výskytů těchto dat v daném intervalu odpovídajících absolutní nebo relativní četností. Histogram je vizuální pomůckou, která přehledně informuje o rozdělení četností statistických termínů. To můžeme říci, že je nejužívanější cesta k vyjádření statistických dat. Pro vytváření histogramu je třeba brát existující vstupní data, která jsou umístěna v jednom sloupci, v něm na buňku můžeme napsat název. A druhý sloupec obsahuje čísla tříd, která představují intervaly, podle kterých chceme měřit frekvenci.

4.3.6 Konstrukce

Představíme si, daný soubor dat, v nichž jsou hodnoty, které se rozmísťují podél číselné osy. Když chceme vytvořit histogramy, tak na této ose nakreslíme prvky nebo body v rozděleném intervalu. Na obrázku níže jsou data rozdělena do pěti intervalu vodorovné osy, šířka v intervalech musí být stejná. Dále spočítáme, kolik objektů je v daném intervalu a z těchto počtů objektů nakreslíme obdélník (bar) dle požadované velikosti. Jeho výška pak odpovídá počtu datových bodů. Do svislé osy nakreslíme výšky dle počtu datových bodů v každém intervalu a označíme popisek dat v této ose. A tímto postupem právě vytvoříme histogram. S histogramy můžeme pracovat i samplování, podle určeného rozsahu v ose zadáváme počet vzorkování do určitého intervalu a vygenerujeme histogram v přehledu pro výstupní objekty.

Povaha histogramu závisí na rozsahu dat, který je umístěn v intervalu. Můžeme usnadňovat pohyb histogramu. Představíme si nějaký příklad. Skupina sportovců skórovala v každém inter-

valu hodnot od 100 do 600. Každé histogramy, které jsou načteny od distribuce datové sady, v kterých jsou ukládány data výsledků od kategorie sportovce, jsou v jednotlivém intervalu a má vlastní výšku histogramu. Můžeme přetáhnout histogram stojící v intervalu 500 a přidat ho do intervalu 400 a tím získáme měřené hodnoty ve společném intervalu 400 a 600. Histogram ukazuje, jakou velikost datové sady spadá do rozmezí. Další možnost, že histogram zobrazuje takový tvar přibližující distribuci dat. Často, když vytvoříme histogram z datové sady, můžeme porovnat teoretické rozdělení. Například skóre jsou navržena tak, aby odpovídaly běžné distribuci dat, nazývá se normální rozdělení nebo Gaussovo rozdělení. Původní histogram bývá normální rozdělení, protože je schopný vytvářet další intervaly v rámci stejného rozsahu hodnot. S růstem velikosti vzorku můžeme rozdělit histogramy na tenčí intervaly a jeho tvar se dostane blíž k normálnímu rozdělení.

4.3.7 Normalizace

Tvar histogramu nám říká, že frekvence jako hodnoty na ose y (svislá osa) je užitečná ve specializovaných případech. Změna hodnoty na ose y beze změny histogramu je známá jako normalizace. A může se provést několika různými způsoby.

- Pravděpodobnostní rozdělení je takové pravidlo, které každému jevu přiřazuje pravděpodobnost. Někdy je dobré vědět, že 12 sportovců dosáhlo skóre mezi 300 a 350, a to je často užitečné vědět, jaká pravděpodobnost je při náhodném výběru sportovce, který obdrží body od 300 do 350. Jinými slovy chceme změnit rozložení četnosti do diskrétní rozdělení pravděpodobnosti. V něm je součet všech výšek histogramu pravděpodobnosti 1 (nebo 100%). Převod z frekvence na pravděpodobnosti můžeme provést tak, že rozdělíme každou danou frekvenci celkové velikosti vzorku. Například daný interval s frekvencí 50 z celkového počtu 200 datových bodů se stává pravděpodobnost 0,25 nebo 25%. Vzhledem k tomu, že vydělíme každou část histogramu stejným číslem, všechno se zmenšuje ve stejném poměru a tvar zůstává stejný.

- Hustota pravděpodobnosti používá k popisu rozdělení spojité náhodné veličiny místo pravděpodobnostní funkce. Je to reálná nezáporná funkce. Můžeme mít v úvahu tuto každou pravděpodobnost a rozdělit ji na šířku intervalu (nikoliv změnit celkový tvar), poté převedeme své diskrétní rozdělení pravděpodobnosti na rozložení hustoty pravděpodobnosti. Tyto histogramy se používají k modelování funkce hustoty pravděpodobnosti, které mají vlastnosti, že v oblasti umožňují sloučit různé hodnoty histogramu v různých intervalech dané náhodné proměnné. To znamená, že oblast pod celým histogramem se rovná 1, protože podle definice hustoty pravděpodobnosti je 1. Funkce hustoty pravděpodobnosti pro pevný rozsah dat je dobrý způsob, jak porovnáme histogramy různých samplování. (zvětšuje velikost vzorku, intervaly se tenčí, výšky zůstanou srovnatelné)

- Hustota frekvence je rozložení četností, které je rozdělenou šířkou rozsahu v intervalu. V této oblasti je součet k celkovému počtu datových bodů ve vzorku.

Následující tabulky ukážou příklad vyplňující data. Tento symbol # v prvním řádku a sloupci nám říká, že je vazba mezi těmito dvěma tabulkami. Pro přehled typů normalizace jsme uvedli vzorky pro histogram, který je označen n jako datový bod a Λx je délka intervalu. V tomto příkladu je celkový počet datových bodů $(N) = 15$ a $(\Lambda x) = 100$.

Tabulka 1: Typ histogramu a jeho formule

#	**Typ histogramu**	**Formule**
1	Rozdělení četností	n
2	Diskrétní rozdělení pravděpodobnosti	$\frac{n}{N}$
3	Diskrétní procentuální rozdělení pravděpodobnosti	$\frac{100n}{N}$
4	Hustota frekvence	$\frac{n}{\Lambda x}$
5	Hustota pravděpodobnosti	$\frac{n/N}{\Lambda x}$

Tabulka 2: Frekvence velikosti vzorku

#	**100 - 200**	**200 - 300**	**300 - 400**	**400 - 500**	**500 - 600**
1	1	4	5	3	2
2	0,067	0,267	0,333	0,200	0,133
3	6,67	26,7	33,3	20	13,3
4	0,01	0,04	0,05	0,03	0,02
5	6,66e-4	2,67e-3	3,33e-3	2e-3	1,33e-3

Popis dle pořadí #

1. Součet počtu datových bodů (jeden bar = datový bod)
2. Součet baru, který spadá do pravděpodobnosti
3. Součet baru, který spadá do formy procentuální pravděpodobnosti
4. Součet baru a délky v intervalu = hustota v rámečku
5. Součet baru a délky v intervalu, který spadá do pravděpodobnosti

4.3.8 Příklad

Můžeme uvést příklad ke konstrukci samplování. Pokud máme k dispozici nějaká vstupní data malé populace, tak je distribuce samplování znázorněna $(0, 2, 4, 6, 8)$.

Populace: $[0, 2, 4, 6, 8]\mu = 4; \sigma = 2,828$

Opakované vzorkování s náhradou za různé velikosti vzorků prokáže, že lze vyrobit různé distribuce samplování. Distribuce samplování je závislá na velikosti vzorků. Jako příklad s vzorky o velikosti dvou bychom navrhovali číslo, řekněme 6 a z toho je pravděpodobnost 1 v 5, tak se rovná 0,2 nebo můžeme psát 20 %. Potom toto číslo dáme zpět a nakreslíme další číslo. Řekněme, že se jedná o číslo 8. Z toho vychází průměr (střední hodnota) v našem vzorku $N = 2$ je nyní 7,

protože $(6+8)/2 = 7$. Nyní se můžeme již místo vylosovaného čísla věnovat zpět k populaci. σ je vypočítán pomocí N ve jmenovateli, spíše toto označení je standardní než $N-1$, protože máme populaci a ne vzorky. Různé hodnoty pro vzorkování znamenají pro každou velikost vzorku včetně společně s pravděpodobnosti výskytu každé střední hodnoty. Můžeme očekávat třeba klasicky 30 vzorků. Například, pro $N = 2$, je jediný způsob, jak dostaneme střední hodnotu s nulou, je-li první a druhý odběr nuly. Z tohoto důvodu je pravděpodobnost $\frac{1}{5} \times \frac{1}{5} = \frac{1}{25} = 0,04$. Pro 30 odběrů bychom očekávali toto $\frac{1}{25} \times 30 = \frac{30}{25} = 1,2$.

Tabulka 3: Velikost samplování. X - počet vzorku, P - pravděpodobnost, O - očekávaní

N = 2			N = 3			N = 4		
X	P	O	X	P	O	X	P	O
0	0,04	1,2	0	0,008	0,24	0	0,0016	0.048
1	0,08	2,4	0,67	0,024	0,72	0,5	0,0064	0,192
2	0,12	3,6	1,33	0,048	1,44	1,0	0,0160	0,480
3	0,16	4,8	2,00	0,080	2,40	1,5	0,0320	0,960
4	0,20	6,0	2,67	0,120	3,60	2,0	0,0560	1,680
5	0,16	4,8	3,33	0,144	4,32	2,5	0,0832	2,496
6	0,12	3,6	4,00	0,152	4,56	3,0	0,1088	3,264
7	0,08	2,4	4,67	0,144	4,32	3,5	0,1280	3,840
8	0,04	1,2	5,33	0,120	3,60	4,0	0,1360	4,080
	1	**30**	6,00	0,080	2,40	4,5	0,1280	3,840
			6,67	0,048	1,44	5,0	0,1088	3,264
			7,33	0,024	0,72	5,5	0,0832	2,496
			8,00	0,008	0,24	6,0	0,0560	1,680
				1	**30**	6,5	0,0320	0,960
						7,0	0,0160	0,480
						7,5	0,0064	0,192
						8,0	0,0016	0,048
							1	**30**

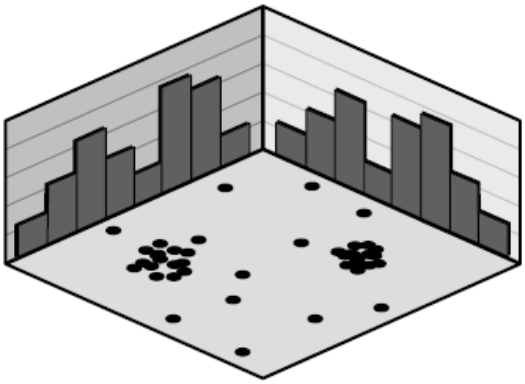
Jak je známo grafů, střední hodnota μ základního souboru je 4 a směrodatná odchylka σ je 2.828. U vzorkování o velikosti $N = 2, 3, 4$ s největší pravděpodobností hodnoty pro X v rozdělení výběru vzorků je 4,0 (tato hodnota je pro μ). Fakt, že se to tak stane pro statistiku, u které nazýváme střední hodnoty výběru tak, jak vyvoláváme myšlenku, že střední hodnotou tohoto výběru je nestranný odhad populace v základním souboru. Jinými slovy na základě průměru ve vzorku dané velikosti můžeme očekávat, že střední hodnota výběru se může rovnat hodnotě populace (to nemusí vzhledem k jedné nebo druhé hodnotě populace). Pro konkretní vzorek můžeme do určité míry vynechat střední hodnotu s odhadem (buď příliš vysoká nebo příliš nízká), existuje jiná varianta, jak získáme odhad populace na distribuci samplování rozptyl a směrodatnou odchylku.

5 Vícerozměrný histogram

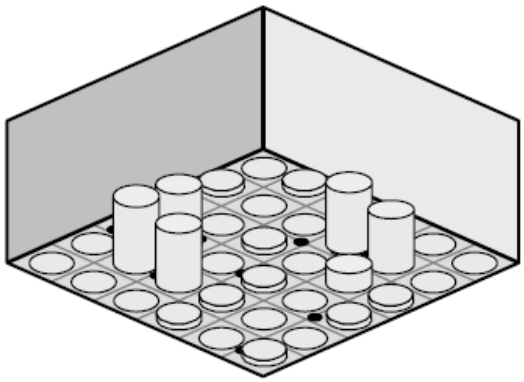
Vícerozměrné histogramy představují použití jednoduchého statistického údaje, například minimální a maximální hodnoty v daném sloupci pro odhad selektivity vícerozměrného dotazu. Pomocí těchto jednoduchých statistik se bude produkovat odhad k selektivitě pouze tehdy, když jsou rovnoměrně rozděleny hodnoty atributů. Vzhledem k tomu, že hodnoty atributů mohou mít jiné rozložení, to se stalo samozřejmostí pro relační optimalizaci dotazu používající histogramů pro odhadování selektivity faktorů. Nicméně tyto histogramy mají tradičně stejnou šířku. Equiwidth histogramy podle citace autora [26] mohou způsobit chybné odhady selektivity, pokud nejsou rovnoměrně rozloženy hodnoty atributů. Podle této citace taktéž v oblasti tohoto problému sestavení histogramů na jediný atribut byl dobře prostudován. Bylo taktéž prokázáno, že způsob, jak kontroluje maximální odhad chyby, provádí kontroly hloubky každého histogramu nikoliv jeho šířky. Jinými slovy, všechny třídy histogramu musí mít stejnou šířku nikoli stejnou četnost. Je třeba řadit vztah na konkrétní atribut ke generování hloubky histogramů. Maximální chybu odhadu selektivity lze libovolně snížit zvýšením počtu hloubky třídy.

5.1 Algoritmus při generování

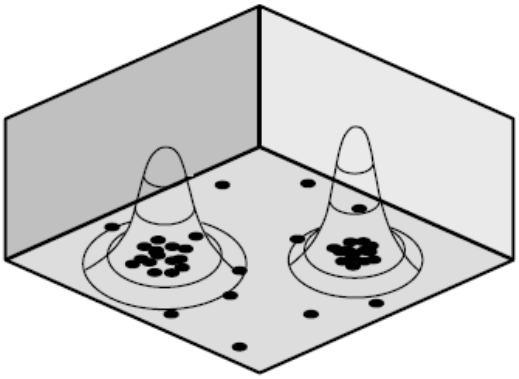
Jak vypadají vícerozměrné histogramy? Předpokládáme, že máme algoritmus pro generování vícerozměrných histogramů, ale nejprve můžeme trochu popsat situaci k vytváření těchto histogramů. Představujeme si příklad s dvěma dimenzemi, ve kterých leží relace R s atributy x a y. Obrázek 5 ukáže takový obdélník ABCD, který představuje prostor v relaci R. Uvnitř tohoto obdélníku leží body. Tento obdélník se nazývá hloubka histogramu nebo třída histogramu. Budeme dále používat termíny třídy a histogram zaměnitelně. Počet třídy požadovaným výrazem $S = bucket_1 * bucket_2$. Bucket$_i$ se používá k označení počtu rozdělení podél atributu (dimenze). Proto počet bodů v každé třídě $= \frac{N}{S}$, kde N je celkový počet bodů. Pro zjednodušení následující vysvětlení, budeme předpokládat, že $= \frac{N}{S}$ je integrální počet. Technika pro generování histogramu obsahuje tři parametry. První parametr je číslo atributu, na kterém vztah má být seřazen vzestupně. Druhý a třetí parametr je nízký respektive vysoký. Tato dvě slova jsou sériová čísla dvou záznamů ve vztahu tak, že kolekce v rozmezí od nízké až k vysoké jsou seřazeny na atributu daném prvním parametrem. První celý vztah (1 až N), se nejprve třídí na první atribut. Pak se tvoří $bucket_1$ oddíly stejné velikosti. První oddíl se skládá 1 až $\frac{N}{bucket_1}$. Druhý oddíl se skládá z tvaru v podobě n-tice $\frac{N}{bucket_1} + 1$ až $2 * \frac{N}{bucket_1}$. Tyto tvary se jednají o primární oddíly. Pak se třídí každý z těchto primárních oddílů na druhém atributu a pak se rozdělí každý primární oddíl na $bucket_2$ sekundární oddíly. Důležité je, že sekundární oddíly, které jsou vytvořeny z jediného primárního oddílu, jsou zcela uzavřeny v rámci těchto nadřízených primárních oddílů. Tak se vytvoří počet sekundárních oddílů pro tyto formy ($bucket_1 * bucket_2$), z nichž každý obsahuje tvar dané n-tice $\frac{N}{bucket_1 * bucket_2}$ [14]. Každý z těchto sekundárních oddílů odpovídá třídě a naopak. Každá třída může být zastoupen souřadnicemi, jeho levý dolní a pravý horní roh. V levé spodní části x, y souřadnice třídy je jednoduše nejnižší hodnota prvního (druhého) atributu z bodů



(a) Jednorozměrné histogramy

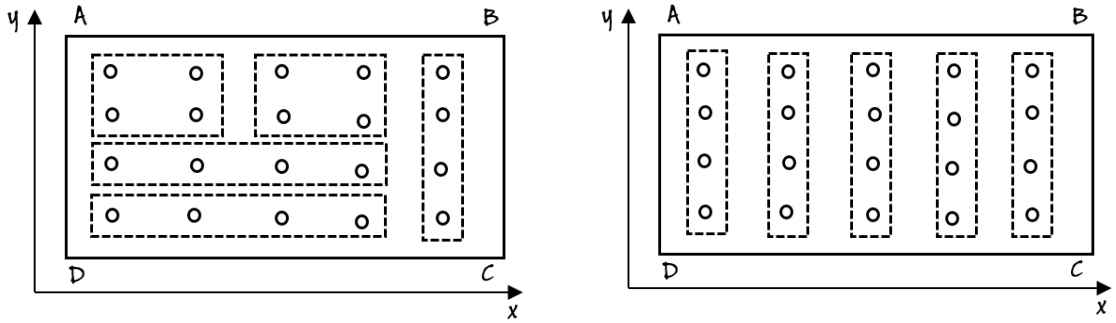


(b) Vícerozměrné histogramy



(c) Odhad selektivity prostřednictví shlukování

Obrázek 4: Vizualizace různých konceptů pro odhad selektivity [27].



Obrázek 5: Dva rozdílné řešení s 5 bucket, každé bucket má 4 body

v příslušném sekundárním oddílu. Podobně v pravé horní části x, y souřadnice třídy je nejvyšší hodnota prvního (druhého) atributu z bodů v příslušném sekundárním oddílu.

Pokud rozšiřujeme do tří dimenzí, tak potřebujeme vyřešit třídění každého ($bucket_1 * bucket_2$) sekundárního oddílu na třetím atributu a rozdělení každého sekundárního oddílu do $bucket_3$ pro terciární oddíly. Opět platí, že každý z terciárních oddílů je zcela uzavřený v rámci nadřazeného sekundárního oddílu. Pak máme celkem ($bucket_1 * bucket_2 * bucket_3$) počtu oddílů v rámci přísné hierarchie, každý má stejné počty bodů. Každý z těchto oddílů terciárních odpovídá 3-dimenzionální třídě [14].

Každý objekt z reálného světa můžeme popsat jeho pozicí v mnohorozměrném prostoru. Tento problém u vícerozměrné analýzy se snaží řešit různými přístupy. Tyto druhy přístupů nabízejí redukci dimensionality dat sloučením korelovaných proměnných do menšího počtu proměnných a identifikují shluky předmětů a následné zmenšení více-dimenzionálního problému v kategorizaci objektů do zjištěných shluků. Vícerozměrná analýza nám pomáhá nalézt nejvhodnější pohled na data poskytující maximální informace o analyzovaných objektech v různém dimenzionálním prostoru. Podle této citace [4] v kapitole 3 se inspirujeme, jak se vygenerují multidimenzionální histogramy. Dále popisujeme podrobnosti následujících metod pro nalezení odhadů této selektivity.

5.2 GENHIST

Je statický odhad selektivity. Oddíl GENHIST (pro zkratku v angličtině GENeralized HISTograms) identifikuje husté části v souboru dat a přiřazuje jim buckets. Pak se odstraní datové body a začne se znovu. GENHIST poskytuje dobré odhady selektivity, ale jsou vysoké náklady při jeho vytváření a provozu. Problém je v tom, že všechny statické histogramy mohou být pravidelně přestavěny tak, aby odrážely změny v datovém souboru. Představíme techniku algoritmu GENHIST nalezení vícerozměrného histogramu pro datové sady z reálné domény. Při řešení rozsáhlého odhadu selektivity můžeme používat tuto metodu. Jádro odhadu je zobecněním samplování. Jako vzorkování nalezení výsledku u odhadů je tento postup efektivní a dostatečný, průhledný průchod pro multidimenzionální data [11].

Dle této příručky [12], která byla provedena výsledkem z experimentálního pokusu, je tato technika je schopna účinně postavit selektivitu odhadů pro multidimenzionální datové sady s reálnými atributy. Při testování se zdá, že GENHIST je robustní a přesná technika. Má dobrou konkurenceschopnost pro přesnost odhadů z vícerozměrného jádra. Nejenom to, ale má i tu výhodu, že může být vypočítána průchodem v jedné datové sadě (stejně jako výběr vzorků). Nicméně, funguje lépe než výběr vzorků pro dimensionality. Je to důvodně ukázáno, že je volbou pro výpočet výsledku v závislosti na rychlost. Obtížné je vybrat správné hodnoty v různých parametrech.

Pomocí této techniky se vytváří statický histogram, který se skládá z obdélníkového bucket. Obrázek 9 je ukázka histogramu techniky GENHIST. Histogram se vytváří pomocí následujícího heuristického přístupu - při zvýšení pravidelné mřížky od dané nejmenší velikosti se vybere daný

počet buněk v nejhustší mřížce tak, aby se každý z nich převedl do bucket. Výsledek histogramu může obsahovat překrývající bucket. GENHIST nevyužije zpětnou vazbu dotazu a je obtížné vyladit více parametrů jako původní mezery v mřížce, počet segmentů vytvořených v každé iteraci a rychlosti, s níž roztečí zvyšuje regiony v prostoru pro každou iteraci [12].

5.3 Equi-Depth

Multidimenzionální Equi-Depth histogramy [27] je technika, která umožňuje identifikovat jádro, které má nejhustší shluky. Histogramy se rozdělí na prostor do bucket s proměnlivou velikostí a pevným bodem frekvence. Její algoritmus pro konstrukci vícerozměrných histogramů equi-depth je zobrazeno tak, jak datový prostor iteračně rozdělí podél každého požadovaného atributu na pevný počet bucket, kde pořadí atributů je pevné. Selektivita po dotazu q je odhadován analogicky jako jednodimenzionální histogramy užívající bucket v uváhu, že se s q protínají. Obrázek 9

5.4 STHoles

STHoles je takový přístup, který navrhuje hierarchicky organizované vícerozměrné histogramy. Algoritmus STHoles je vícerozměrný dynamický histogram, který vygeneruje libovolné rozložení bucket s omezením, jeho bucket se nemusejí překrývat, jak je znázorněno na obrázku 9. Bucket umožňuje obsadit v jiných bucket a tím vytvoří strukturu „rodič-potomek“ v hierarchii. Součást hierarchie histogramů je reprezentována jako strom, kde každý uzel představí bucket. Pomocí tohoto hierarchického pojetí a nerovnoměrného rozložení uvnitř bucket b přijímají přesněji několik menší bucket uvnitř b. STHoles histogramy jsou konstruovány s použitím sady ukázkových dotazů jako reference. Regiony v datovém prostoru, které jsou dotazovány častěji, tedy mohou být zobrazeny podrobněji prostřednictvím většího počtu bucket. Histogramy jsou rafinované po každém dotazu. Postup se však stará, že ne více než stanovená horní mez bucket je generována. Je-li tato horní mez porušena dočasně během reorganizace, bucket jsou sloučeny [22].

„Self-Tuning“ je odhad selektivity. Zkrátka self-tuning histograms se vyjadřuje ST histograms [28]. Poukazuje na to, že je schopen řešit problém pomocí zatížení dotazu. To vede k histogramům, které jsou citlivější na pracovní vytížení dotazu. Nyní stručně popisujeme „state-of-the-art“ odhad selektivity techniky STHoles. Obecně platí,že odhad selektivity se pokusí odhadnout počet objektů, který splňuje predikát dotazu.

Nyní popisujeme obecnou strukturu STHoles histogramu. Následující podkapitolou představíme definice a konstrukce histogramu pro algoritmy na výstavbu nových histogramů.

5.4.1 Definice histogramu

Jak již bylo vysvětleno v předchozí části, vztah zařazení mezi bucket poskytuje další stupeň flexibility ve srovnání s režimy, které využívají dělení disjunktní bucket. Každý bucket b v STHoles histogramu se skládá z obdélníkového ohraničujícího rámečku označující $box(b)$, reálná hodnota

frekvence označující $f(b)$, ve které se ukáže největší počet bodů v ohraničení bucket. V tradičním histogramu bucket b je oblast, která je považována za jednotnou hustoty n-tice. Ten bucket může mít otvory, které jsou samy o sobě prvotřídním histogramem bucket. Tyto otvory jsou pro bucket potomek a jejich ohraničovací rámečky jsou disjunktní a jsou zcela uzavřeny v b ohraničujícího pole[1]. Z tohoto důvodu STHoles histogram může být koncepčně viděn jako stromová struktura, kde každý uzel představuje bucket.

S ohledem na histogram H přes datovou sadu D a rozsah na dotazu q, odhadovaný počet v D, který leží uvnitř q. Můžeme označit podobu $odhad(H, q)$ je:

$$odhad(H, q) = \sum_{b \in H} f(b) \frac{v(q \cap b)}{v(b)}$$

kde $v(q \cap b)$ označujeme objem průniku q a b. V následujícím odstavci představíme algoritmy k vybudování a vylepšení STHoles histogramů.

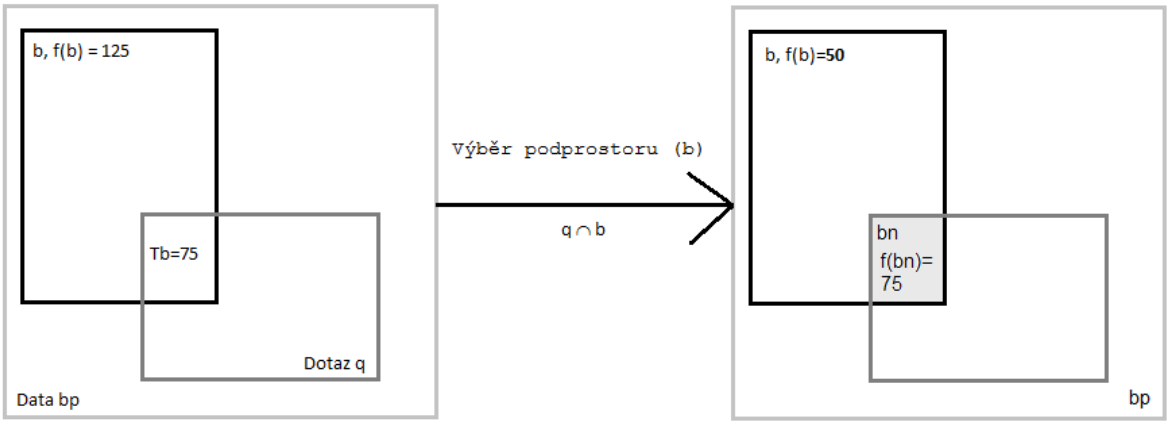
5.4.2 Konstrukce histogramu

Hlavní myšlenkou vybudování histogramů STHoles je zachycení výsledku dotazů v pracovním zatížení a účinné shromaždiště několika jednoduchých statistik nad nimi, postupné vylepšení rozvržení a četnost existujících bucket. Regiony, které jsou více dotazovány, mohou mít tímto způsobem prospěch z více bucket s jemnější granularitou. Jestli chceme vytvořit STHoles histogram, tak začneme s prázdným histogramem neobsahující žádný bucket. Alternativní řešení je, pokud máme více informací o distribuci dat, např. z celkového počtu záznamů v datovém souboru s maximální a minimální hodnotou pro každý atribut[2], tak můžeme začít s jedním histogramem z bucket. Pro jejich experimenty budeme předpokládat, že neznáme hodnoty z datové sady, a proto začínáme s prázdným histogramem. Jinak obecněji můžeme použít existující histogram a začít s přesnějším modelem datovém souboru.

Po nastavení počátečního histogramu pro každý dotaz q v pracovní zátěži se ukáže výsledek toku a my spočítáme, kolik je bodů uvnitř každého segmentu v aktuálním histogramu. Pokud aktuální dotaz q přesahuje hranice v kořenovém bucket, tak se rozšiřuje ohraničení rámečku kořenového bucket tak, aby tento dotaz q pokryl. Regiony v oblasti dat, které mohou využít podle této sekce 5.4.3 a upřesnění informace na „výběr podprostoru“, nebo zvětšování v bucket, které pokrývá oblast dotazu 5.4.4. V kapitole výsledný histogram se pomocí sloučení podobné bucket konsolidovat 5.4.5.

[1] Alternativní provedení může přidat frekvenci do bucket potomků na frekvenci správného bucket. Tento alternativní design vyjadřuje přesně stejné informace jako tyto STHoles histogramy.

[2] I když přibližný celkový počet n-tic v souboru dat můžeme získat z katalogů systému, mohou být náklady na údržbu v nepřítomnosti indexů maximální a minimální hodnoty pro každou vlastnost.



Obrázek 6: Zlepšení přesnost histogramu

5.4.3 Identifikace kandidátských výběrů

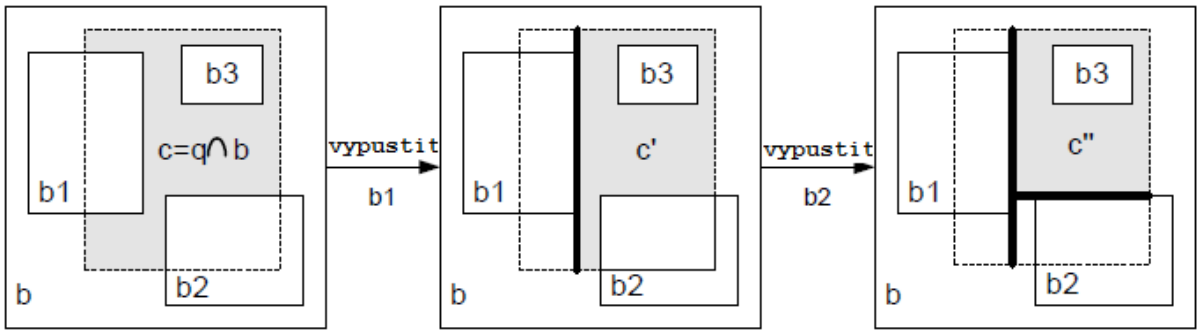
V této sekci ukážeme postup k využití identifikace výběru v bucket se STHoles histogramem výsledku na dotazu q. Tyto výběry, které odpovídají subregionům bucket s rozlišovacími prvky frekvence, využívají se k zpřesnění STHoles histogramu.

Obecně platí, že dotaz q kříží některé bucket pouze částečně. Pro každý bucket b_i známe přesný počet n-tice prvků za $q \cap b_i$ s přezkoumáním výsledku dotazu q. Pokud $q \cap b_i$ má neúměrně velký nebo malý zlomek v bucket b_i, pak $q \cap b_i$ je kandidátem na díru bucket b_i. Proto každý dílčí průnik q a histogram v zásadě může použít ke kvalitě histogramu, jak je znázorněno v příkladu níže.

Pokud průsečík dotazu q a bucket b je obdélníkový, jak je na obrázku 6, může se vždy považovat $q \cap b_i$ jako kandidát na výběr podprostoru a proces probíhá jako následující příklad. Tento například z výsledného toku pro dotaz leží n-tice bodů $Tb = 75$ v části bucket b, který se dotýká dotazu q, $q \cap b$. Můžeme zlepšit přesnost histogramu, pokud se vytvoří nový bucket bn pomocí „výběr“ podprostoru v b, který odpovídá průniku $q \cap b$, upraví b a bn a aktualizace frekvence dle způsobeného výpočtu. V podstatě se zmenší $q \cap b_i$ ve velké obdélníkové podoblasti, která se částečně neprotíná s ohraničením rámečku jiného bucket. Technologie odhadu předpokládá, že počet n-tice v tomto subregionu je jednotnost. To znamená, že v případě T_b počet n-tice za $q \cap b_i$ a c je výsledkem smršťování $q \cap b_i$. T_c je odhadem za počet n-tice v c, jako $T_c = T_b \frac{v(c)}{v(q \cap b)}$.

Postup pro smršťování (**`Shrink`**) průsečíku v bucket b a dotazu q je v článku [20].

Popsali a identifikovali jsme pro každý dotaz q nové kandidující výběry, které zdokonalí daný histogram.



Obrázek 7: Smrštění výběrového podprostoru $c = q \cap b$

5.4.4 Výběr kandidáta jako nový histogram

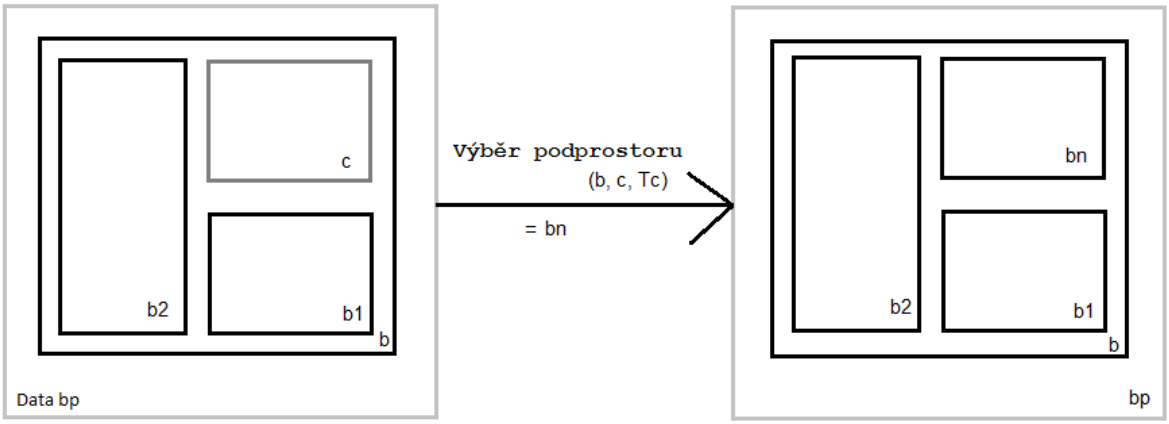
Předchozí sekce ukázala, jak identifikovat výběr kandidáta nového výběrového podprostoru pro STHoles histogram. Každý kandidát na výběr podprostoru c s frekvenci T_c, který vyplývá ze smršťování $q \cap b_i$. To je zcela zahrnuto v b_i a částečně se neprotíná s žádným potomkem B_i. Jak je znázorněno na obrázku 7. Nyní můžeme ukázat, jak se efektivně „výběr podprostoru“ takoví zájemci nového histogramu v bucket. Za tímto účelem můžeme identifikovat tři možné scénáře:

1. Bucket b_i a výběr podprostoru c je v doméně tzv. $box(c) = box(b_i)$. Výběr kandidáta nese aktualizované informace o počtu n-tice bodů v bucket b_i, T_c (frekvence), ale nevybíráme podprostor c v b_i, protože představuje v podstatě stejnou oblast. Podle této situace existuje řešení za nahrazení frekvence b_i s T_c.

2. Kandidátský výběr podprostoru c pokrývá všechny bucket b_i v prostoru. To je poměrně vzácný speciální případ, ale potřebujeme to zpracovat správně tak, aby se zabránilo plýtvání prostorem. Když se díváme na obrázek 8 u histogramu se čtyřmi bucket b_1, b_2, b a b_p, předpokládáme, že chceme vybrat podprostor c v bucket b. Přestože $c \neq box(b)$, c pokryje všechny zbývající prostory b (zbytek je krytý bucket b_1 a b_2). Pokud bychom jednoduše přidali nové podřízené b_n na bucket b s $box(b_n = c)$, pak řádný bucket b by nenesl žádnou informaci, protože b by byl naprosto s krytý jeho podřízeným b_1, b_2 a b. Proto přidáním b_n jako nové části podřízené b, by mělo za následek zbytečný prostor. Aby se předešlo této situaci, budeme eliminovat bucket b z histogramu a přeneseme část bucket b jako „potomek“ (children) do b_p „rodič“ (parent). Konkrétně se spojí b s jeho nadřízeným bucket b_p a pak vybereme znovu c, ale tentokrát v b_p, čímž se šetří jeden bucket, který stojí v prostoru [3].

3. Výchozí situace. Myšlenky od začátku kapitoly 5.4.2 můžeme přímo aplikovat. To znamená, že jsme vytvořili nového potomka b_i, který se označí jako b_n s $box(b_n) = c$ a $f(b_n) = T_c$. Pak jsme migrovali všechny potomky b_i, jejichž ohraničující boxy jsou kompletně zahrnuty c, takže se potomek stává novým bucket b_n.

[3] Jako alternativu můžeme zabránit slučování b a b_p a potom vrtáme b_n s jednoduchou změnou frekvence b. Však výsledky jsou menší přesahy mezi bucket, které jsou obecně žádoucí.



Obrázek 8: Výběr b_n v bucket b by nesl žádnou užitečnou informaci

5.4.5 Spojení buckets

Předchozí oddíl ukázal, jak můžeme vylepšit STHoles histogram pomocí přidání výběru podprostoru bucket do existující bucket. Přitom můžeme dočasně překročit náš cílový počet histogram bucket. Proto po přidání bucket bychom měli snížit počet histogramu bucket sloučením podobné skupiny, konkrétně bucket s nejbližší hustotou n-tice. Obecně řečeno s rozhodnutím, že bucket, který chceme sloučit, tak použijeme penále, která vrátí náklady v histogramu přesnosti sloučení oba buckety.

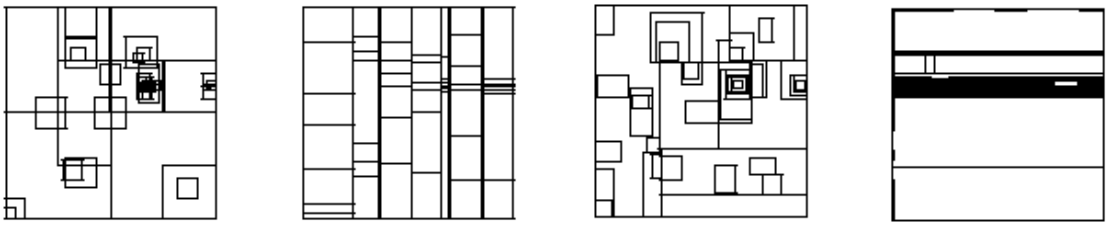
5.5 MLGF

Tato metoda představuje dynamickou hashovací organizaci souboru, která umožňuje přístup k souboru s více atributy. Výraz MLGF je vlastně anglickou zkratkou (multilevel grid file). MLGF představuje rozšíření souboru navržené podle tohoto autora [23]. Toto rozšíření řeší řadu nevýhod v souboru, která způsobuje jeho multidimenzionální adresářové pole.

5.6 MHIST

Je statický odhad selektivity. Rekurzivně rozdělí datové sady, počínaje jedním bucket reprezentující celé sady dat. V každé iteraci rozdělí stávající bucket na dva a tak až do vyčerpání uloženého prostoru. Bucket se rozdělí a rozdělení dimenze je zvolena nenasytně na základě jednorozměrných histogramů. Multidimenzionální souborový histogram atributu je konstruován rozdělením společné distribuce dat do vzájemně disjunktních bucket, sbližování frekvence a jeho hodnotu nastavuje v každém segmentu jednotným způsobem. Funguje dobře pro vysoce šikmé distribuce dat. Dokonce se věnuje příliš mnoho bucket v nejhustších shlucích [4].

Hodnota v doméně je rozšířeně aproximována na jednotné šíření. Nechť nejmenší a největší hodnoty X_i v bucket B jsou min_i a resp. max_i. Pak si můžeme představit tento bucket jako



(a) GenHist (b) EquiDepth (c) STHoles (d) MHist

Obrázek 9: Rozložení bucket v histogramu

n-rozměrný obdélník s dvěma krajními zákoutími. Nechť d_i je počet různých hodnot v atributu X_i, pak jsou přítomny v B. Ať k je přibližná hodnota v dimenzi (jednotné rozpětí by se získalo za předpokladu podél této dimenze), označí se $v'_i(k)$. Aktuální bodová data v B jsou pak aproximovat všech možných kombinací $< v'_1(k_1), \ldots, v'_n(k_n) >$, kde $1 \leq k_i \leq d_i$. Zde na obrázku 9

Všechny histogramy jsou vytvořeny z jednotné frekvence a aproximují v bucket průměrné seskupené frekvence. Například, je-li F je součtem všech frekvencí v bucket B, tak každá orientační hodnota je spojena s přibližnou frekvencí $F/(d_1 \times .. \times d_n)$.

6 Singulární rozklad

6.1 Předmluva

Singulární rozklad (SVD) je široce používanou metodou pro rozklad matice do několika dílčích matic, vystavující mnoho užitečných a zajímavých vlastností původní matice. Rozklad matice se často nazývá faktorizace - rozložení čísla na součin menších čísel. V ideálním případě matice se rozloží do řady faktorů (často ortogonální nebo nezávislé), které jsou optimální na základě nějakého kritéria. Toto kritérium může být například rekonstrukce rozložené matice. Rozklad matice je také užitečný, když není plně hodnost matice. Hodnost matice je menší nebo rovna menšímu z čísel řádků a sloupců. To znamená, že řádky nebo sloupce matice jsou lineárně závislé. Teoreticky lze použít Gaussovu eliminační metodu ke snížení řádku v matici na trojúhelníkový tvar a pak počet nenulových řádků, u kterých se navzájem určí hodnost. Avšak tento přístup není praktický při práci v aritmetice s konečnou přesností. Podobný případ se prezentuje při použití LU rozkladu, kde L je dolní trojúhelníková matice s jednotkami na diagonále a U je horní části matice v trojúhelníkovém tvaru s prvky na diagonále. V případě nedostateku hodnosti matice může rozložit do menšího počtu faktorů, než původní matice, a přesto zachová všechny informace v matici. SVD obecně představuje rozšíření původních dat v souřadnicovém systému, ve kterém kovarianční matice je diagonální.

Pomocí SVD je možné určit dimenze v rozsahu matice. Hodnost matice je rovena počtu lineárně nezávislých řádků nebo sloupců. SVD může také kvantifikovat citlivost lineárního systému pro numerické chyby nebo získání inverzní matice. Navíc poskytuje řešení problémů a zpracovává situaci, kdy matice jsou buď singulární nebo číselné obory velmi blízké k singulární.

SVD se využívá zejména při zpracování signálů a obrazů (např. konstrukce, obnova nebo komprese obrazu) pro biomedicínské inženýrství (např. využití při analýze DNA), teorie řízení a systémů v mnohých oblastech inženýrství(např. řešení homogenních lineárních rovnic, statistika, vyhledávače). V následující sekci ukážeme aplikace, které se používají v těchto oblastech, vyžadují hodnost matice A, třídní matice, ortonormální báze a projekce. Dálšími spolehlivými nástroje k výpočtu za přítomnosti hodnot jsou singulární čísla a singulární vektory.

SVD je ortogonální transformace, kterou lze optimálně zachytit základní varianci dat. Tato vlastnost je velmi užitečná pro snížení dimenzionaliyu v multidimenzonální kolekci a pro podporu smysluplné vizualizace dat. SVD má řádu významných aplikací kromě redukce dat. Patří mezi ně inverzní matice, komprese dat a přičtení neznámých datových hodnot.

Dále SVD umožňuje obrázkovou kompresi dat. Komprese dat řeší důležitou aplikaci zejména pomocí lineární algebry. V moderním světě stále roste přenášení dat v podobě digitálních informací, je třeba tato množství minimalizovat. Singulární rozklad je účinným nástrojem pro minimalizaci ukládání a přenosu dat. Tato prezentace se zabývá kompresí obrázků pomocí singulárního rozkladu na obrazové matici.

SVD obsahuje tři vzájemně kompatibilních hlediska (úhly pohledu). Na jedné straně vidíme jako metodu transformaci korelovaných proměnných do nekorelované sady, které umožňuje lépe odhalit různé vztahy mezi původními datovými položkami. SVD je metoda pro identifikaci a pořadí dimenze, podél které datové body vykazují největší rozdíly. To se váže na třetí sledování SVD, což je možné zjistit, kde je většina variant, najít nejlepší aproximace (přiblížení) původních datových bodů s použitím méně rozměrů. Proto lze považovat SVD za metodu pro redukci dat. Regresní přímka, která prochází skrze body, se ukáže na aproximaci původních dat s jednodimenzionálním objektem (čárou). V tom smyslu, kde je aproximace, je linka, která minimalizuje vzdálenost mezi každým původním místem a čárou.

Singulární rozklad ihned odhaluje několik maticových vlastností včetně hodnosti, čísla podmíněnosti a důležitého zobrazení o struktuře matice, jako je součin tří matic $X = USV^T$, kde

- U matice je složena sadou z levé pozice na ortonormální bázi
- S matice je diagonální matice
- V matice je složena sadou z pravé pozice na ortonormální bázi

Hodnoty v S hrají hlavní roli v singulární funkci, nazývají se singulární hodnoty. Jsou kladné (nezáporné) a jejich veličiny ukazují význam odpovídajících bází (komponenty).

SVD v podstatě provádí otáčení souřadnic, které sladí transformované osy se směrech maximální odchylky v datech. To je užitečný postup za předpokladu, že pozorovaná data mají vysoký poměr signálu k šumu, a že velký rozptyl odpovídá zajímavému obsahu dat, zatímco nižší rozptyl odpovídá hluku. Používáme dolní index vektoru odkazující na komponent v této pozici.

6.2 Vektor

Vektory jsou obvykle označovány malým písmenem se šipkou nahoru $\vec{x}$. Čísla obsahující vektor se nazývají komponenty a počet komponentů se rovná dimensionalitu vektoru.

$$\vec{x} = \begin{pmatrix} 3 \\ 6 \\ 8 \\ 5 \\ 7 \end{pmatrix}$$

Například $\vec{x}$ je 5-dimenzionální vektor, $x_1 = 3, x_2 = 6, x_3 = 8, x_4 = 5, x_5 = 7$. Vektory mohou reprezentovat horizontální tvar pro úsporu místa, např. $\vec{x} = [3, 6, 8, 5, 7]$ je ekvivalentní vektor jako je uvedeno výše. Obecněji vektor s n-dimensonality je vlastně posloupnost n čísel, a komponent představuje hodnotu ve vektoru $\vec{x}$ o velikosti dimenze i^n.

Tabulka 4: Mezinárodní soutěž Fitness. Datový zdroj: Muscle and Fitness červenec 1997

Soutěžící	Kolo 1	Kolo 2	Kolo 3	Kolo 4	Celkem	Pořadí
Závodník 1	17	18	5	5	45	1
Závodník 2	42	28	30	15	115	3
Závodník 3	10	10	10	21	51	2
Závodník 4	28	5	65	39	132	5
Závodník 5	24	26	45	21	116	4

6.3 Matice

Matice je tabulka dat. Ukázka pro tuto tabulku 4, která ukazuje pět nejlepších střelců v pořadí na ukazatele výkonnostních měření u organizace mezinárodní soutěže v roce 1997. Tabulka 4 se skládá z řádků[4] a sloupců[5]. To, co vidíme v tabulce, je typ matice obdélníkové. Pole obsahující čísla v této tabulce zapíšeme do tvaru matice, potom matice vypadá takto:

$$\begin{bmatrix} 17 & 18 & 5 & 5 & 45 & 1 \\ 42 & 28 & 30 & 15 & 115 & 3 \\ 10 & 10 & 10 & 21 & 51 & 2 \\ 28 & 5 & 65 & 39 & 132 & 5 \\ 24 & 26 & 45 & 21 & 116 & 4 \end{bmatrix}$$

Velikost nebo dimenze matice je dána počtem řádků a sloupců v tabulce. Tím vznikne matice v rozměru 5×6. Můžeme pojmenovat popis pro proměnnou hodnotu, ve které je skryté reálné číslo a tradičně označujeme symbol matice A. Maximální počet řádků je přiřazen proměnné m a počet sloupců se pojmenuje n. Prvky[6] v matici jsou označovány malým písmenem a, jeho index řádku je označený malým písmenem i a sloupce j. Uvedeme malý příklad, vybereme třeba nejvyšší hodnotu v matici 132. Její pozice je v řádku 4 a sloupci 5. Můžeme ji nazývat $a_{45} = 132$. Obecněji řečeno, popis pro pozici řádku i a sloupce j, je tedy a_{ij}.

Třída matice

Než začneme rozebírat třídu matice do podrobnosti, je dobré mít znalost jejího matematického pojmu. Matice je tabulka o m řádcích a n sloupcích. Pak hovoříme o typu mxn. V každé buňce této tabulky je jiné číslo, jiný výraz a různá hodnota. Dále mohou mít matice různé vlastnosti. Některé speciální matice mají dokonce vlastní názvy např. čtvercová matice.

Nech $m, n \in \mathbb{R}$ $(i = 1, \ldots, m, j = 1, \ldots, n)$. Potom matice typu $m \times n$ nazýváme následující

[4] horizontální seznam výsledků, v kterém je účast soutěžícího.
[5] vertikální seznam čísel ze skóre pro dané kolo.
[6] můžeme také nazývat položky, pole nebo komponenty

tabulkou

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

Začínáme $A = (a_{ij})_{m \times n}$, zkráceně $A = (a_{ij})$. Budeme pracovat jen s maticemi nad polem $\mathbb{R}$. Prvky matice jsou označeny indexy udávajícími řádek a sloupec, v nichž se prvek ve skutečnosti nalézá. Prvek v i-tém řádku a j-tém sloupci matice A se někdy značí a_{ij}. To znamená, že pak i-tý řádek matice obsahuje vodorovnou n-tici prvků $(a_{i1}, a_{i2}, \ldots, a_{in})$, kde $i = 1, 2, \ldots, m$ a j-tý sloupec matice obsahuje svislou m-tici čísel $(a_{1j}, a_{2j}, \ldots, a_{mj})$, kde $j = 1, 2, \ldots, n$.

S matici můžeme provádět operaci nad vektory. Matice je v podstatě sbírkou vektorů. Můžeme pak mluvit o vektorech řádku nebo vektorech sloupce. Nebo dokonce vektor s n komponenty a potom považujeme za označení matice $1 \times n$.

6.4 Vektorové terminologie

Vektorová délka

Délka vektoru se zjistí srovnáním jednotlivých komponent. Je-li $\vec{v}$ vektor, tak jeho délka je označena $|\vec{v}|$. Přesněji,

$$|\vec{v}| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

Pro příklad, je-li $\vec{v} = [4, 11, 8, 10]$, pak

$$|\vec{v}| = \sqrt{4^2 + 11^2 + 8^2 + 10^2} = \sqrt{301} = 17,35$$

Sčítání vektorů

Vložení dvou vektorů se rozumí přidání jednotlivé komponenty v $\vec{v_1}$ do komponenty v příslušné pozici $\vec{v_2}$ a získá se nový vektor $\vec{v_3}$. Například $\vec{v_1} = [3, 2, 1, -2]$, $\vec{v_2} = [2, -1, 4, 1]$

$$[3, 2, 1, -2] + [2, -1, 4, 1] = [(3 + 2), (2 - 1), (1 + 4), (-2 + 1)] = [5, 1, 5, -1]$$

Obecněji, je-li $A = [a_1, a_2, \ldots, a_n]$ a $B = [b_1, b_2, \ldots, b_n]$, pak

$$A + B = [a_1 + b_1, a_2 + b_2, \ldots, a_n + b_n]$$

Násobení vektorů

Násobení vektoru skalárem (reálným číslem) lze geometricky reprezentovat jeho prodloužením nebo zkrácením, popřípadě změnou jeho orientace na opačnou (při násobení záporným číslem). Například je-li $\vec{v} = [3, 6, 8, 4]$, pak $\frac{3}{2} * \vec{v} = \frac{3}{2} * [3, 6, 8, 4] = [\frac{9}{2}, 9, 12, 6]$ Obecněji, je-li d reálné číslo a $\vec{v}[v_1, v_2, \ldots, v_n]$ je vektor, pak $d * \vec{v} = [dv_1, dv_2, \ldots, dv_n]$.

Skalární součin

Skalární součin dvou vektorů definuje násobení vektorů mezi sebou. Výsledkem skalárního součinu dvou vektorů je číslo. Skalární součin je definován pouze pro vektory stejného rozměru (dimenzi). Skalární součin dvou vektorů je označován $(\vec{v_1}, \vec{v_2})$ nebo $\vec{v_1} \cdot \vec{v_2}$ (skalární součin). Tedy,

$$(\vec{x}, \vec{y}) = \vec{x} \cdot \vec{y} = \sum_{i=1}^{n} x_i y_i$$

Například, je-li $\vec{x} = [1, 6, 7, 4]$ a $\vec{y} = [3, 2, 8, 3]$, pak

$$\vec{x} \cdot \vec{y} = 1(3) + 6(2) + 7(8) + 3(4) = 83$$

Ortogonalita

Dva polohové vektory v rovině či prostoru jsou navzájem ortogonální, pokud jejich skalární součin se rovná nule. Vektory jsou kolmé v dvourozměrném prostoru nebo mezi nimi je úhel 90°. Například vektory $[2, 1, -2, 4]$ a $[3, -6, 4, 2]$ jsou ortogonální, protože

$$[2, 1, -2, 4] \cdot [3, -6, 4, 2] = 2(3) + 1(-6) - 2(4) + 4(2) = 0$$

Jednotkový vektor

Jednotkový[7] vektor je vektor, jehož se rovná v délce jedné. Každý vektor s počáteční délkou > 0, může být normalizovaný podíl jednotlivé komponenty podle délky vektoru. Například je-li $\vec{v} = [2, 4, 1, 2]$, pak

$$|\vec{v}| = \sqrt{2^2 + 4^2 + 1^2 + 2^2} = \sqrt{25} = 5$$

Potom $\vec{u} = [\frac{2}{5}, \frac{4}{5}, \frac{1}{5}, \frac{2}{5}]$ je jednotkový vektor, protože

$$|\vec{u}| = \sqrt{\left(\frac{2}{5}\right)^2 + \left(\frac{4}{5}\right)^2 + \left(\frac{1}{5}\right)^2 + \left(\frac{2}{5}\right)^2} = \sqrt{\frac{25}{25}} = 1$$

[7] nebo můžeme říct normálový vektor

Ortonormalita

Dva vektory $\vec{u}$ a $\vec{v}$, které mají jednotkovou délku, jsou-li ortogonální, pak jsou ortonormální. Například $\vec{u} = \left[\frac{2}{5}, \frac{1}{5}, -\frac{2}{5}, \frac{4}{5}\right]$ a $\vec{v} = \left[\frac{3}{\sqrt{65}}, -\frac{6}{\sqrt{65}}, \frac{4}{\sqrt{65}}, \frac{2}{\sqrt{65}}\right]$ jsou ortonormální, protože

$$|\vec{u}| = \sqrt{\left(\frac{2}{5}\right)^2 + \left(\frac{1}{5}\right)^2 + \left(-\frac{2}{5}\right)^2 + \left(\frac{4}{5}\right)^2} = 1$$

$$|\vec{v}| = \sqrt{\left(\frac{3}{\sqrt{65}}\right)^2 + \left(-\frac{6}{\sqrt{65}}\right)^2 + \left(\frac{4}{\sqrt{65}}\right)^2 + \left(\frac{2}{\sqrt{65}}\right)^2} = 1$$

$$\vec{u} \cdot \vec{v} = \frac{6}{5\sqrt{65}} - \frac{6}{5\sqrt{65}} + \frac{8}{5\sqrt{65}} - \frac{8}{5\sqrt{65}} = 0$$

Gramova-Schmidtova ortogonalizace

Gramův–Schmidtův ortogonalizační proces je metoda pro konverzi vektorů do ortonormálních vektorů. Například tento sloupec vektorů

$$A = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 2 & 0 \\ 2 & 3 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

převedeme konverzi do ortonormálních vektorů

$$A = \begin{bmatrix} \frac{\sqrt{6}}{6} & \frac{\sqrt{2}}{6} & \frac{2}{3} \\ 0 & \frac{2\sqrt{2}}{3} & \frac{-1}{3} \\ \frac{\sqrt{6}}{3} & 0 & 0 \\ \frac{\sqrt{6}}{6} & \frac{\sqrt{-2}}{6} & \frac{-2}{3} \end{bmatrix}$$

první normalizace vektoru $\vec{v_1} = [1, 0, 2, 1]$:

$$\vec{u_1} = \left[\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right]$$

Dále, nechť

$$\vec{w_2} = \vec{v_2} - \vec{u_1} \cdot \vec{v_2} \cdot \vec{u_1} = [2, 2, 3, 1] - \left[\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right] \cdot [2, 2, 3, 1] \cdot \left[\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right]$$

$$= [2, 2, 3, 1] - \left(\frac{9}{\sqrt{6}}\right) \cdot \left[\frac{1}{\sqrt{6}}, 0, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right]$$

$$= [2, 2, 3, 1] - \left[\frac{3}{2}, 0, 3, \frac{3}{2}\right]$$

$$= \left[\frac{1}{2}, 2, 0, -\frac{1}{2}\right]$$

Získáme normalizaci vektoru $\vec{w_2}$

$$\vec{u_2} = \left[\frac{\sqrt{2}}{6}, \frac{2\sqrt{2}}{3}, 0, \frac{-\sqrt{2}}{6}\right]$$

Nyní spočítáme vektor $\vec{u_3}$ z hlediska vektorů $\vec{u_1}$ a $\vec{u_2}$ takto,

$$\vec{w_3} = \vec{v_3} - \vec{u_1} \cdot \vec{v_3} \cdot \vec{u_1} - \vec{u_2} \cdot \vec{v_3} \cdot \vec{u_2} = \left[\frac{4}{9}, -\frac{2}{9}, 0, -\frac{4}{9}\right]$$

a normalizace vektoru $\vec{w_3}$

$$\vec{u_3} = \left[\frac{2}{3}, -\frac{1}{3}, 0, -\frac{2}{3}\right]$$

Obecněji, pokud máme ortonormální vektory $\vec{u_1}, \ldots, \vec{u_{k-1}}$, pak $\vec{w_k}$ je vyjádřen jako

$$\vec{w_k} = \vec{v_k} - \sum_{i=1}^{k-1} \vec{u_i} \cdot \vec{v_k} \cdot u_i$$

6.5 Maticové terminologie

Diagonální matice

Diagonální maticí A nazýváme čtvercovou matici $n \times n$, u které prvky na hlavní diagonále mají různé hodnoty od nuly a všechny ostatní prvky rovny nule. Jinými slovy pouze nenulové hodnoty běží podél hlavního dialogu od levého horního rohu do pravého dolního rohu. Někdy se zapisuje jako $diag(-1, 1, 1, 1)$

$$A = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Čtvercová matice

Je taková matice, která má stejný počet řádků a sloupců. Pro příklad, jestli třeba určíme velikost čtvercové matice s n řádky a sloupci, tak volíme stejnou velikost n-čtvercové. Například tabulka matice s 3-čtvercovou velikostí.

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

6.6 Ortogonální matice

Matice A je ortogonální jestliže $AA^T = A^TA = I$. (jednotková matice)

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{5} & \frac{-4}{5} \\ 0 & frac45 & frac35 \end{bmatrix}$$

je ortogonální, protože

$$AA^T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{5} & \frac{-4}{5} \\ 0 & frac45 & frac35 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{3}{5} & \frac{4}{5} \\ 0 & frac-45 & frac35 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Jednotková matice

Jednotková matice s velikosti n je čtvercová matice $n \times n$, která má na hlavní diagonále jedničky a nuly na ostatních místech. Jednotková matice se chová jako číslo 1 v běžné multiplikační matici, což znamená, že $AI = A$, ukážeme si malé příklady k její charakteristice.

$$I_1 = \begin{bmatrix} 1 \end{bmatrix}, I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \ldots, I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Jednotková matice je speciálním případem diagonální matice.

$$A = \begin{bmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \end{bmatrix} I = \begin{bmatrix} 1 & 0 & \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 4 & 6 \\ 1 & 3 & 5 \end{bmatrix}$$

Transpozice matice

Matice, která byla vytvořena z matice A. Provede se vzájemně výměna řádků a sloupců, označujeme to jako transponovanou matici a značíme A^T. To znamená, že řádek 1 se stane sloupcem 1, řádek 2 bude sloupec 2, atd. Například, pokud transponovaná matice bude

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

pak její transpozice je

$$A^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

Multiplikační matice

Dvě matice se mohou násobit jen tehdy, pokud má druhá matice má počet řádků, který je shodný s počtem sloupců první matice. Výsledná matice po výpočtu má tolik řádků jako první matice a tolik sloupců jako druhá matice. Jinak řečeno, pokud matice A je $m \times n$ a matice B je $n \times s$, pak výsledek matice AB je $m \times s$. Souřadnice AB jsou určeny tím, že skalární součin každého řádku v matici A a každého sloupce v matici B. To znamená, že je-li $A_1, \ldots, A_m$ jsou řádky vektoru v matici A, a $B^1, \ldots, B^8$ jsou sloupce vektoru v matici B, pak ab_ik v matici AB se rovná $A_i \cdot B^k$. Například níže.

$$A = \begin{bmatrix} 2 & 1 & 4 \\ 1 & 5 & 2 \end{bmatrix} \qquad B = \begin{bmatrix} 3 & 2 \\ -1 & 4 \\ 1 & 2 \end{bmatrix}$$

$$AB = \begin{bmatrix} 2 & 1 & 4 \\ 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 3 & 2 \\ -1 & 4 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 9 & 16 \\ 0 & 26 \end{bmatrix}$$

$$ab_{11} = \begin{bmatrix} 2 & 1 & 4 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix} = 2(3) + 1(-1) + 4(1) = 9$$

$$ab_{12} = \begin{bmatrix} 2 & 1 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} = 2(2) + 1(4) + 4(2) = 16$$

$$ab_{21} = \begin{bmatrix} 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \\ 1 \end{bmatrix} = 1(3) + 5(-1) + 2(1) = 0$$

$$ab_{22} = \begin{bmatrix} 1 & 5 & 2 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} = 1(2) + 5(4) + 2(2) = 26$$

Determinant

Determinant je funkce v čtvercové matici, která je schopna převést hodnoty na jedno číslo. Tento

determinant v matici A je označen $|A|$ nebo $det(A)$. Jestliže matice A obsahuje ve tvaru 2 x 2, pak

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc$$

Pro příklad, determinant matice

$$A = \begin{bmatrix} 4 & 1 \\ 1 & 2 \end{bmatrix} pak |A| = \begin{vmatrix} 4 & 1 \\ 1 & 2 \end{vmatrix} = 4(2) - 1(1) = 7$$

Nalezení determinantu matice, která má čtvercový tvar, $n > 2$, lze rekurzivně odstranit jeden řádek a sloupce pak vytvořit postupně menší matice tak, dokud se nedostane na čtvercový rozměr v matici 2×2. Determinant matice se zmenší každý odpovídající odstraněný řádek a sloupec. Postup:

$$A = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix}$$

- Vynásobí se determinant matice 2×2, která na pozici a není v řádku nebo sloupci
- Stejně tak pro pozici b a c
- Přidat odebranou hodnotu ale nesmíme zapomenout na pozici b, která má znaménko minus

$$|A| = a \begin{vmatrix} e & f \\ h & i \end{vmatrix} - b \begin{vmatrix} d & f \\ g & i \end{vmatrix} + c \begin{vmatrix} d & e \\ g & h \end{vmatrix}$$

Příklad:

$$A = \begin{bmatrix} 6 & 1 & 1 \\ 4 & -2 & 5 \\ 2 & 8 & 7 \end{bmatrix}$$

$|A| = 6(-2 \cdot 7 - 5 \cdot 8) - 1(4 \cdot 7 - 5 \cdot 2) + 1(4 \cdot 8 - (-2 \cdot 2)) = 6(-54) - 1(18) + 1(36) = -306$

6.7 Odhad

Jak je popsáno, výše uvedené podkapitoly základního úvodu pro výpočet SVD. Jako názorový příklad vezmeme trojici matici s označením U a V. Z toho vyplývá, že můžeme udělat v X libovolně faktorizaci. Pro vstupy (záznamy) U a V bývají hodnoty vetšinou mezi 1 a -1. Taková možnost faktorizace je možná.

Pro dvou-dimenzionální matice M, nechť $R_M(i)$ je horizontální vektor odpovídající i-tému řádku M a $C_M(i)$ je vertikální vektor odpovídající i-tému sloupce M.

$$X = \sum_{k=1}^{N} d_k C_u(k) R_v(k)$$

Z dvou-dimenzionální matice vyplývá, že pro odhady velikosti výsledku může být vypočítán její jednotný vektor a jednotné hodnoty na základě definovaného singulárního rozkladu. Toto pozorování motivuje právě použití metody SVD, která indukuje datové informace z distribuční sítě.

Technika

Zvažme společná distribuční data Γ o dva atributy X_1 a X_2 (dvou-dimenzionální vlastnosti) s jejich hodnot velikosti D_1, D_2 ($D_1 \geq D_2$). Nechť M je matice odpovídající společné frekvenci. Pak Γ je orientační na základě následujících kroků:

1. Výpočet SVD dle postupu zdrojových kódů pro jazyka C [24].
2. Pro číslo $k <= N$, ukládá přesně odpovídající nejvyšší singulární hodnoty k.
3. Vytváření jedno-dimenzionálních histogramů na $2k$ řádkových a sloupcových vektorech (matic) odpovídá těmito podmínkám. V zásadě lze odlišnost od taxonomie histogramu lze použít pro každý vektor, ale v praxi má smysl používat jeden stejný pro všechny z nich.

Postavení histogramu podle kroku 3 může být zapojeno do vzorce $X = \sum_{k=1}^{N} d_k C_u(k) R_v(k)$, na tomto místě řádkových a sloupcových vektorů získá přiblížení X a následně požadované společné distribuce dat. V kroku 2 je rozpracováno tak, aby bylo zřejmě, že když mohutnost atributů bude vyšší (vysoká N), tak velikost matice bude také velmi vysoká, proto řešení aproximace všech singulárních vektorů je poměrně nepraktická. Ukazuje se, že když atributy vysoce závislé nebo téměř nezávislé, tak distribuce pro diagonální hodnotu d_i bývá velmi zkosená (několik vysokých a většinou velmi malé hodnoty) [25]. Jako výsledek ke skladování histogramů lze získat pouze první k termínu (výrazu) SVD, jeden může dostat poměrně dobré přiblížení ve společné frekvenci matice. Máme na mysli konkrétní instance SVD na bázi aproximace pomocí k termínů jako označení SVD-k techniky.

V poslední době se často využívá několik užitečných metod pro multidimenzionální odhady selektivity podle autora, který navrhl techniku [4]. Získané společné distribuce dat rozdělíme do disjunktních buckets, tedy do tří matice (U, D, V). Diagonální matice D závisí na velikosti rozsahu diagonálních zápisů. Dále U jsou levé singulární vektory a V jsou pravé singulární vektory. Tyto vektory jsou singulárně rozdělené pomocí libovolného jednorozměrného histogramu tak, aby bylo použito jako histogram buckets atributů. Existuje mnoho účinných SVD algoritmů, ale metodu SVD lze použít pouze u dvou rozměrů.

7 Implementace a výsledky experimentů

7.1 Visual Studio

Pro implementaci této aplikace na metodu odhadu jsem použil Visual Studio a C#, z tohoto důvodu, že nabízí různé nástroje, které jsem použil. Visual Studio je integrované vývojové prostředí (IDE) od společnosti Microsoft. Používá se k vývoji počítačových programů pro systém Microsoft Windows, stejně jako webové stránky, webové aplikace a webové služby. Visual Studio softwarové společnosti Microsoft využívá vývojových platforem, jako je rozhraní pro aplikaci programu (API), která je k dispozici v operačních systémech Microsoft Windows. Dále je grafické uživatelské rozhraní (GUI), které umožňuje ovládat počítač pomocí interaktivních grafických ovládacích prvků, zahrnující součást Microsoft .NET Framework, který poskytuje platformu pro zápis bohaté klientské aplikace pro desktop, laptop, tablet i počítač. Nabízí i další grafický subsystém (WPF) pro vykreslování uživatelských rozhraní v aplikacích založených na systému Windows. Umožňuje i vytvářet projekt pro Windows Store, to je hlavním prostředkem šíření aplikace ve stylu Metro a Silverlight, který má aplikační framework pro psaní a spouštění bohaté internetové aplikace, podobně jako aplikace Adobe Flash. Prostředí Visual Studio může produkovat jak nativní kód tak spravovaný kód [29].

Visual Studio podporuje různé programovací jazyky a umožňuje editovat kódu v editoru a podporující debugger v různé míře, téměř zvládne jakýkoli programovací jazyk, do které patří i například C#, pro nástroje v prostředí Visual Stuido je to Visual C#.

V implementační části lze použít toto prostředí Visual Studio a jeho programovací jazyk bude C# podporující GUI v aplikaci (Windows Forms). Tento jazyk je schopen vytvářet aplikaci pro metodu odhadu velikosti.

7.2 Knihovna STHoles

V prostředí Visual C# byla vytvořena vlastní knihovna STHoles. Třída `STHolesAlgorithm` je hlavním řízením odhadu velikosti výsledku. Tato třída umožňuje vytvářet novou instanci, která inicializuje data z datové sady s požadovaným vybraným typem hodnot.

Tato instance volá po provedení metody odhadu v kódu s požadovanými povinnými parametry. Základní metoda v knihovně nabízí dvě možnosti. Tyto varianty závisí na typu kolekce, která zavazuje data nad reálnými čísly (`Double`) nebo data s celočíselnými hodnotami (`Int32`).

```
STHolesAlgorithm(int maxBucketCount, DataTable data,
double min_intersec_dist)
```

Prvním parametrem se rozumí nastavení na maximální počet bucket (výchozí je 10 pro typ `Double` v rozsahu 0..1 a pro typ `Int32` je doporučeno maximální 15 bucket). Druhý argument shromažďuje data, která pak ukládá do vnitřního objektu klasické vnitřní tabulky `DataTable`.

Poslední parametr určí vstupní hodnotu typu `Double`, který nastavuje možnou minimální vzdálenost pro výpočet více bucket tak, aby z důvodu vstupujících reálných čísel nedošlo k překrývání bucket. Například minimální vzdálenost protínání je 0.000000001. S tím konstruktorem lze pracovat pouze s reálnými hodnotami a jejich interval je <0.0 ... 1.0>. Druhá možnost při volání konstruktoru.

```
STHolesAlgorithm(int maxBucketCount, DataTable data,
int xmax, int ymax)
```

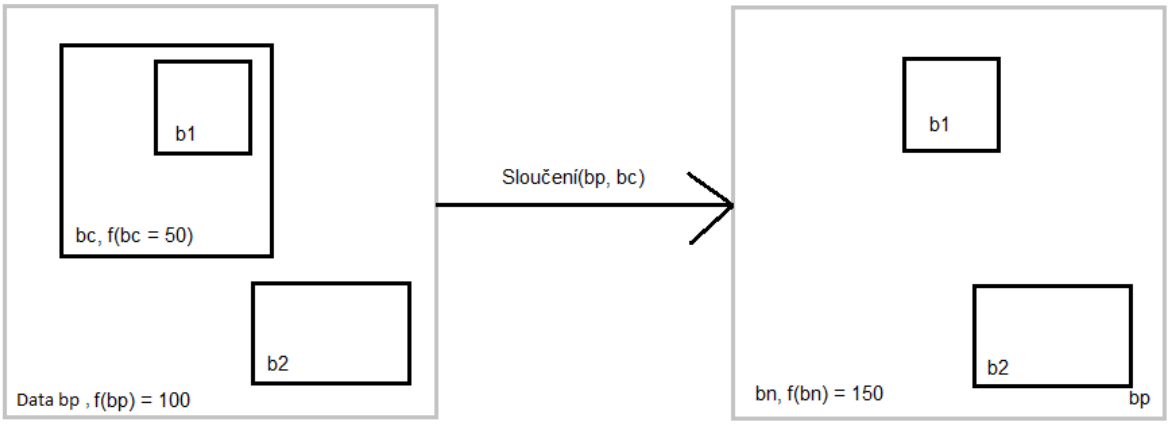
První a druhý argument se prezentuje stejně jak je již popsán výše. Třetí a čtvrtý parametr představuje maximální možnou hodnotu zvolených dat pro osu x a y ve vícerozměrném prostoru.

Hlavní role řízení odhadu je třída `Bucket`. Za vzniknou instanci této třídy je volán konstruktor `STHolesAlgorithm`, který byl předem popsán. Třída `Bucket` umožňuje pracovat s frekvencí (počet bodů ve vícerozměrném prostoru), vytvářet potomky z nadřízeného bucket a ukládat jejich frekvence, vypočítat faktoru objemu, porovnávat objem bucket (šířka a výška), modifikovat hodnoty v jednotlivém bucket, zmenšit nebo zvětšit bucket podle závislosti frekvence a protíná se na základě podmínky, spojení více bucket. Počet bodů v bucket (v implementaci se představuje obdélník) odhalí velikost výsledku podle kroku této citace, kde jsem v oblasti teorie popsal a tak, jak je to popsáno v externí literatuře [20].

Podle této citace jsem implementoval tři body pro vypracování odhadu. Poslední bod pro sloučení bucket má dvě možnosti - rodič a sourozenec. Sloučení bucket se týká té věci, která přesahuje velikost histogramu v jednotlivém bucket. Na základě vypočtu sance vrátí tento objekt, který má dělat buď s vytvořenou novou instanci rodiče nebo sourozence.

1. **Identifikace kandidáta pro výběr a smršťování.**
   `identifyCandidateHoles(...)`
   Tato procedura rozumí tomu, že se snaží identifikovat díru histogramu v mateřském bucket po aktuálním výsledku frekvence na dotazu. Jestli se protíná bucket z dotazu a nadřízený bucket, tak se zrodí na základě kandidáta výběru nový bucket, který aktualizuje účast pro nový histogram. Pokud se stane více účasti, tak se provádí výpočet ke smršťování. Názorná ilustrace je na obrázku výše 6.

2. **Provedení na výběr kandidáta**
   `drillHole(...)`
   Metoda v implementaci výběr kandidáta má tři možnosti. V prvním případě jsou kandidátské bucket, které jsou odkazovány, stejné $box(c) = box(b_i)$. U druhého scénáře kandidátský výběr zahrnuje všechny zbývající prostory. Jedná se o zvláštní případ, měli bychom s tím správně zacházet, aby se zabránilo plýtvání volným prostorem. Můžeme si vzít v úvahu na obrázku 8. b = bucket, c = kandidátská díra, Tc = počet dané frekvence pro c. V tře-



Obrázek 10: Sloučení bucket dle Parent-Child

tím případě je implicitní postavení, na kterém je standardní operace pro řízení bucket. (přesunout se, smazat, přidání, modifikace frekvence).

3. **Sloučení bucket** Hlavní role pro určení sloučení bucket za předpokladu, že chceme sloučit dva bucket $b1$ a $b2$ v daném histogramu H, je výpočet sankcí. H' je výsledný histogram po sloučení. Obecně definujeme sankce pro sloučení b_1 b_2 v daném histogramu H.

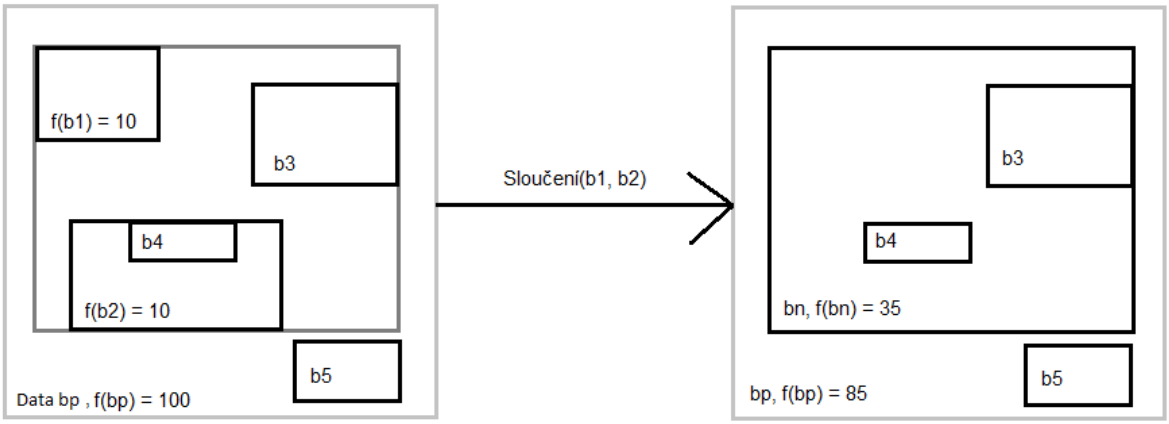
$$penalty(H, b_1, b_2) = \int_p |est(H, p) - est(H', p)|dp$$

   (a) `idetifyBestParentChildMerge()`
   Vrací nejlepší možný společný rodič po výpočtu dané penalty, že bude spojení bucket s druhým bucket jako rodič. Pro výpočet jsem uvedl vzorec, na kterém jsem odkazoval v literatuře.

   Parent-Child Merges
   Předpokládejme, že chceme sloučit bucket bc a bp, kde bp je rodič bc. Po sloučení, obrázek 10, nového bucket bn nahrazující bp, původní bucket bc zmizí. Vzorec k výpočtu pro nový bucket je $box(bn) = box(bn)$ a $f(bn) = f(bc) + f(bp)$. Dva bucket potomků bc a bp se stanou novým bucket bn. Z tohoto důvodu máme vzorec $v(bn) = v(bc) + v(bp)$, který jsem implementoval do prostředí aplikace. Jediné regiony v původním histogramu, které mění odhadovaný počet n-tice bodu po sloučení jsou bp a bc. Závěrem lze říci, že máme výpočet penalty. Tedy:

$$penalty(H, bp, bc) = \left| f(bp) - f(bn)\frac{v(bp)}{v(bn)} \right| + \left| f(bc) - f(bn)\frac{v(bc)}{v(bn)} \right|$$

   (b) `idetifyBestSiblingMerge()`



Obrázek 11: Sloučení bucket dle Sibling-Sibling

Vrací nejlepší možný sourozenec na základě výpočtu sankce. Za sourozence se spojí do jednoho bucket ze dvou bucket.

Sibling-Sibling Merges

Uvažujme sloučení $b1$ a $b2$ se společným mateřským bucket bp, obrázek 11. Nejprve jsme zjistili tmavě šedý rámeček výsledného bucket bn. Definujeme obdélník bn, co nejmenší obdélník, který obklopuje oba bucket $b1$ a $b2$ a neprotíná se s jinými potomkem v celém bucket bp. To znamená, že můžeme začít slučovat ten tmavě šedý rámeček, který těsně uzavře $b1$ a $b2$ a postupně rozšíří, dokud se částečně neprotíná s jinými potomkem v mateřském prostoru bucket bp. Jinak jsme definovali množinu „účastník“ bucket jako soubor potomku bp kromě $b1$ a $b2$, který jsou zahrnuty v $box(bn)$. Po sloučení nového bucket bn nahrazuje bucket $b1$ a $b2$. Obecně může platit, že bn taky bude obsahovat část starého bp. Objem této části je $vold = vBox(bn) - (vBox(b1) + vBox(b2) + \sum_{bi} vBox(bi))$. Proto je frekvence nového bucket $f(bn) = f(b1) + f(b2) + f(bp)\frac{vOld}{vBP}$. Dokonce se může změnit frekvence bp v novém histogramu $f(bp)(1 - \frac{vOld}{vBp})$. Tím se dokončí sloučení bucket v množině účastníka a staří potomci $b1$ a $b2$ se stanou novým potomkem bn. Z tohoto důvodu máme $v(bp) = vold + v(b1) + v(b2)$. Pouze oblasti v původního histogramu, který po sloučení změní odhadovaný počet n-tice bodů, které odpovídají $b1$ a $b2$ i část bp, tmavě šedý obdélník $box(bn)$. Proto:

$$penalty(H, b1, b2) = \left| f(bn)\frac{vOld}{v(bn)} - f(bp)\frac{vOld}{v(bp)} \right| + \left| f(b1) - f(bn)\frac{v(b1)}{v(bn)} \right| +$$

$$+\left|f(b2)-f(bn)\frac{v(b2)}{v(bn)}\right|$$

kde *vOld* je část starého bucket *bp*, na které se vztahuje nový bucket *bn*.

7.3 Objekty

Třída `Bucket` je objekt, který se vytváří na základě vstupní hodnoty obsahu obdélníku. Tento obsah vytvořeného obdélníku je určen pro 2 dimenze, z důvodu této hodnoty jsou v bodech na ose. Vstupní hodnota obsahuje dva body x, y. Na vstupu by byl tvar prvního pole X a druhého sloupce Y. Tato pole a jejich hodnoty jsou vytvořena i v objektu `DataTable`. Bucket je schopen pomocí vstupních proměnlivých hodnot měnit obsah obdélníku. Tvar pro volání konstruktoru je zde.

```
Bucket(Rectangle box, int frequency, Bucket parent, double vFactor)
```

Do instance tohoto objektu se řídí komponent struktury `Rectangle`. Tato struktura je knihovnou standardní Framework. Díky této struktuře získáváme hodnotu šířky a výšky v obdélníku (box). Právě při získávání hodnocení odhadu potřebujeme tyto informace o obsahu boxu. Přijímá a nastaví frekvence podle počtu n-tice bodů v ohraničení bucket. Ověří penále a řídí do větve podle definovaných podmínek na vzorce penalty.

Bucket má několik metod pro řízení rozdělení, spojení, protínání, zmenšení, zvětšení, identifikace výběu kandidáta, aktualizace účastníka v podřízeném bucket, počítaní faktorů v bodech, vrácení výsledků po dotazu, porovnání mezi obsahem obdélníku, nastavení počtu frekvence, atd.

Třída `Query` je objekt, který řídí dotazování na vstupu. Pro 2 dimenze jsou čtyři vstupní hodnoty a mají tvar rozsahových bodů v dotazu jako minimální hodnoty X (zkr. $XMin$), maximální X ($XMax$), minimální Y ($YMin$) a maximální Y ($YMax$).

```
Query(int xMin, int xMax, int yMin, int yMax)
```

Ve vstupním prostředí by tyto hodnoty měly být menší v prvním indexu a následující index pro jeden rozsah v rámci jednotlivého bodu (např. X) by měl být větší. Například vstupní soubor typu csv obsahuje čtyři pole s odděleními $[26, 57, 14, 30]$. První index je pro bod X minimální, druhý index pro X je maximální, třetí pro bod Y je minimální a tak dále.

Třída `Merge` je objekt, který se stará o spojení bucket za podmínky, že zdědí potomka nebo sourozence bucket. Konstruktor této třídy je

```
Merge(double penalty)
```

Tabulka 5: Odhad - střední hodnota s výběrem více než 30 (n > 30)

#	Kolekce	Rozsah hodnoty	Výběrový soubor	Spolehlivost odhadu	Intervalový odhad	Skutečný průměr
1	1000	1..100	200	95%	46,11 ; 53,97	50,24
2	10000	9000..38000	1000	90%	23327,61 ; 24204,83	23399,77
3	25000	0..1	500	90%	0,47 ; 0,52	0,49

Tento objekt, jako parametr, přijímá hodnotu penále a podle toho výpočtu z objektu `Bucket` se zdědí na dvě instance `ParentMerge` (Parent-Child) a `SiblingMerge` (Sibling-Sibling). Tato třída řídí odstranění bucket z mateřského bucket, přidání bucet do nadřízeného bucket nebo přesouvání bucket na sousední bucket.

7.4 Experimenty

Když chceme začít experimentovat odhad, tak můžeme nahrát nejprve vstupní soubor datové sady. Pro svůj experiment jsem použil vlastní testovací prostředí. Tento soubor jsem vygeneroval pomocí náhodných čísel v rozsahu [1 .. 99]. Pak bylo třeba taky vložit do experimentu i dotazy, tak jsem totéž vygeneroval dotazování. Tyto oba soubory doporučujeme typ hodnoty oddělené čárkami (csv). Můžeme experimentovat dva typy metody odhadu **Samplování** a **STHoles**.

7.4.1 Samplování

V případě, že zjišťujeme střední hodnotu základního souboru, mohou nastat dvě situace. Buď známe rozptyl základního souboru (což je obvyklé spíš u poměrně nereálných příkladů), nebo rozptyl základního souboru neznáme a budeme muset použít rozptyl výběrového souboru, který může být zadaný, nebo jej budeme muset vypočítat podle daného vzorce.

Zjišťujeme **střední hodnotu, je-li neznáme rozptyl**. Příklady na výpočet parametrů základního souboru bez toho, abychom znali jeho rozptyl jsou o hodně reálnější, jelikož v realitě opravdu většinou rozptyl základního souboru není známý.

- **Výběr** $n > 30$
  V případě, že neznáme rozptyl, a velikost výběru je větší než 30, můžeme jakékoli rozdělení, kterým se daná náhodná veličina řídí, nahradit normovaným normálním rozdělením. V mé aplikaci se intervalový odhad započítal a máme výsledek v tabulce 5 k dispozici.

- **Výběr** $n < 30$
  Namísto kvantilu normovaného normálního rozdělení se používá **kvantil t Studentova rozdělení**. Tabulka kvantilů Studentova rozdělení má $n - 1$ stupně volnosti.

Zjišťujeme **střední hodnotu, známe rozptyl**. Uvedeme následující příklady, u kterého spočítáme čísla pro odhad. 7

Tabulka 6: Odhad - střední hodnota s výběrem méně než 30 (n < 30)

#	Kolekce	Rozsah hodnoty	Výběrový soubor	Spolehlivost odhadu	Intervalový odhad	Skutečný průměr
1	50	1..5	11	80%	2,39 ; 3,61	2,78
2	6	7000..9000	6	95%	7061 ; 8379	7720

Tabulka 7: Odhad - střední hodnota a rozptyl nebo směrodatná odchylka

#	Rozptyl	Směrodatná odchylka	Průměr	Počet výběru	Spolehlivost odhadu	Intervalový odhad
1	990025		12494	25	95%	12103,96 ; 12884,04
2		7	65	100	95%	63,63 ; 66,37

7.4.2 STHoles

Například v této implementaci knihovny `STholes.dll` jsou umístěny jako interní soubory `RandomDataGenerator.csv` a `RandomQueryGenerator.csv` v interní složce Data, pokud jsou aktivní ovládací prvky Interní u jednotlivého vstupu.

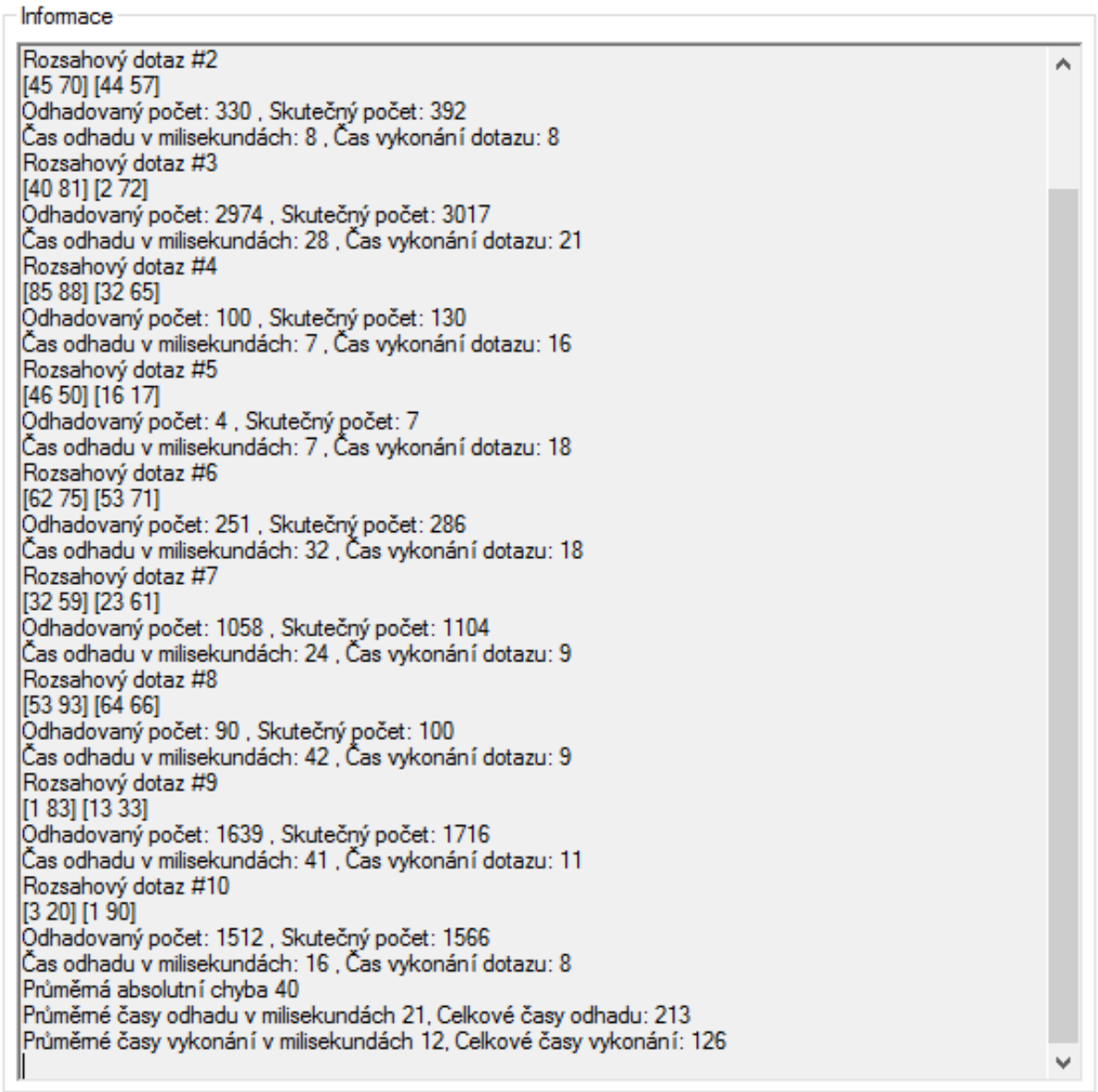
Je zde možnost, že každý uživatel může nahrát vlastní soubory, v tom případě můžeme vypnout tyto prvky a zadat znak oddělovače, který obsahuje oddělovač v externím souboru (to buď čárkou nebo středníkem).12

Jakmile tyto soubory již máme k dispozici a čekají na provedení odhadu, tak můžeme na kliknout tlačítko v rozhraní Odhad. Výsledek pro experiment se ukáže ve skupině tohoto rozhraní Informace na pravé straně. Vložil jsem sekvenci dotazu na 10 kroků. Výsledek je k dispozici na informační zprávu. Zde vidíme, že v informační konzole se vypíšou zprávy o provedení procesu odhadu. Pro každý jednotlivý rozsahový dotaz se ukáže výsledek odhadu a skutečného počtu. Ani nechybí průměrný čas v průběhu na dotazu u odhadu a vykonání dotazu skutečného výsledku. Průměrná absolutní chyba se počítá jako rozdíl mezi odhadem a skutečnosti na obrázku 13.

7.4.3 Vstupní prostředí

- Datové kolekce
  Vstupní datové kolekce byla vygenerována pomocí instance konzole vlastní třídy `RandomDataGenerator` v této implementaci. Při vygenerování kolekce bylo nastaveno 10 tisíc řádků a dvě hodnoty pro osu bylo nastaveno v intervalu [0, 100]. Čím je vyšší počet

Obrázek 12: Uživatelské rozhání pro sekce vstupu.



Obrázek 13: Uživatelské rozhání pro informační zprávu.

Tabulka 8: STHoles: Srovnání mezi odhadem a skutečností pro 10000 záznamů.

Dimenze	2					
Kolekce	10000		**Dotaz**	500 z toho výběru 5		
Rozsah kolekce	1 .. 100		**Typ kolekce**	Celé číslo (Int32)		
Výběr dotazu	**Rozsahový dotaz**	**Odhad**	**Skutečnost**	**Čas odhadu (ms)**	**Čas vykonání dotazu (ms)**	**Relativní chyba (%)**
100	[13;78] [15;61]	3307	3131	0	11	5,6
200	[21;34] [46;77]	487	453	0	8	7,5
300	[15;91] [24;94]	5789	5545	0	14	4,4
400	[72;93] [21;68]	1078	1001	0	9	7,7
500	[9;48] [13;47]	1469	1402	0	10	4,8
Průměrná absolutní chyba pro všechny řádky: 52						

záznamů, tím je lépe zpracován odhad. Můžeme zvýšit počet řádků, ale pro mé testovací prostředí (10 tisíc kolekce) stačí tato hodnota. Po vygenerování se uloží soubor pro příponu csv.

- Rozsahové dotazy
  Vstupní soubor rozsahových dotazů byl taktéž vygenerován pomocí instance vlastní třídy `RandomQueryGenerator`. Tato třída se stará o generování správných vstupních hodnot v rozsahu pro minimální a maximální a za-indexování pole podle dané podmínky. Na 2 dimenzí je nastavení zvoleného počtu řádků pro sekvence dotazů. Přece není možné, aby maximální hodnota byla v prvním indexu a minimální hodnota v druhém poli.

7.4.4 Výsledek jako výstup

V oblasti experimentu můžeme očekávat výsledek jako výstup, který se zpracovává na vstupu, provede se v procesu s řízení bucket a zpracované hodnoty promítají se do výstupní prostředí, které figuruje jako výsledek. Konečný výsledek je v informační zprávě na obrázeku 13. Z tohoto obrázku vyplývá, že proces přijímá vstupní kolekce a na základě počtu řádků adaptuje body na ose v dvourozměrném prostředí. Jakmile kolekce je naučena, tak dalším krokem je pracovat s dotazy pro odhad. V informační zprávě se můžeme dozvědět výsledek nejen odhadu ale i skutečnosti, který byl proveden po jednom řádku dotazu ve vstupním souboru.

V následující tabulce 8 je srovnání odhadu a skutečného výsledku dotazy. Počet kolekce je 10000 a počet dotazu je 500. Vzhledem k tomu, že počet dotazu je přiliš hodně, do tabulky bych to nedával, tak jsem vybral 5 z 500 dotazů. Tento dotaz byl generován rozsah v intervalu 0 až 100. Provedl jsem 500 dotazů a ukážu 5 dotazů z nich v této tabulce 8. Průměr je pro všechny řádky dotazů.

Tabulka 9: STHoles: Srovnání mezi odhadem a skutečností pro milión záznamů.

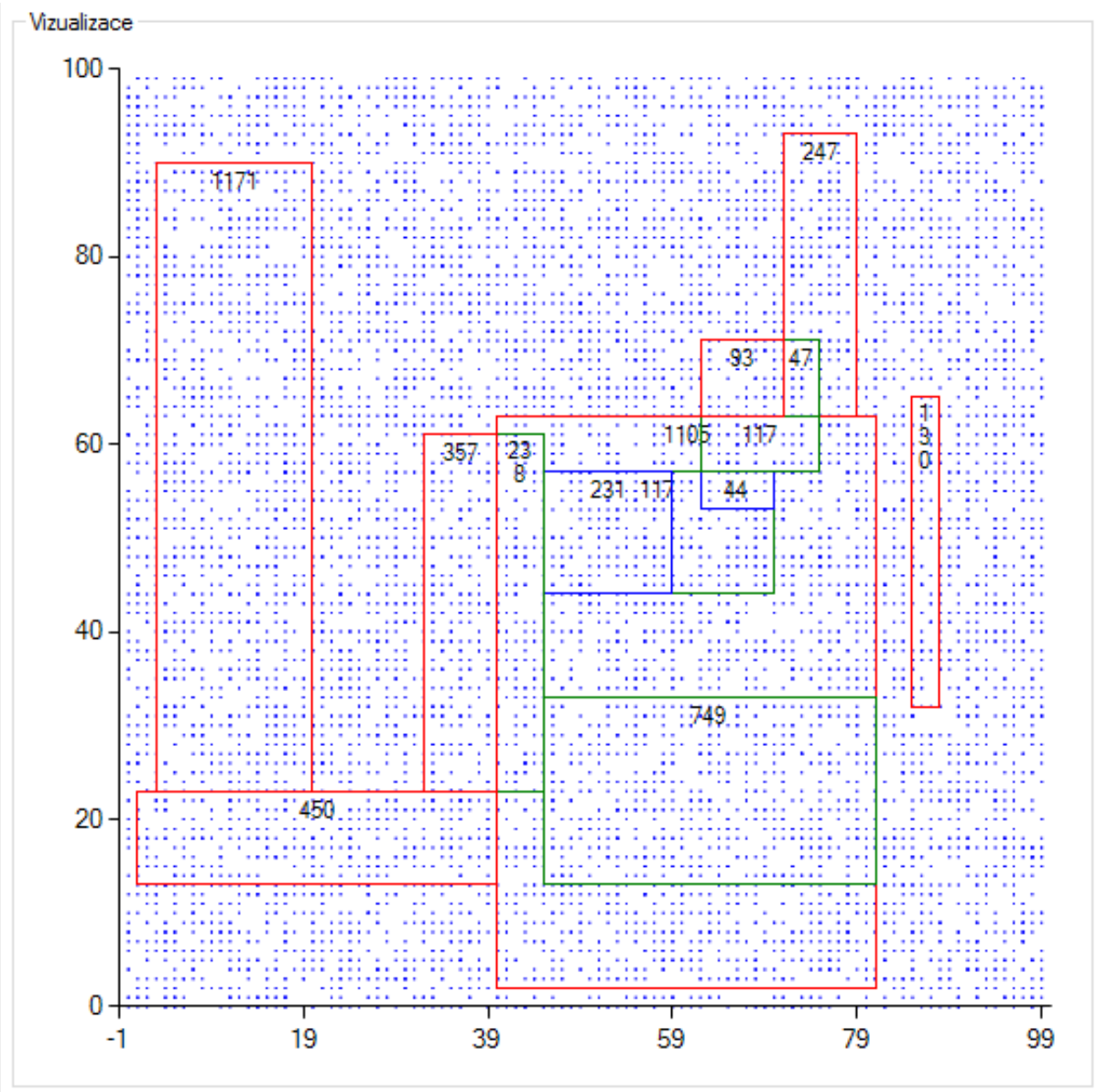
Dimenze	2					
Kolekce	1000000		**Dotaz**	7 z 20000		
Rozsah kolekce	1 .. 500		**Typ kolekce**	Celé číslo (Int32)		
Sekvence dotazu	**Rozsahový dotaz**	**Odhad**	**Skutečnost**	**Čas odhadu (ms)**	**Čas vykonání dotazu (ms)**	**Relativní chyba (%)**
1	[104;345] [207;288]	78397	79132	0	2886	0,93
2	[64;492] [303;451]	254190	256456	0	3372	0,88
3	[113;460] [167;303]	189754	190409	0	3192	0,34
4	[85;198] [117;238]	55070	55302	0	2642	0,42
5	[95;158] [13;340]	83143	83892	0	2584	0,89
6	[214;325] [62;311]	110901	111560	0	2657	0,59
7	[78;372] [291;326]	41971	42314	0	3136	0,81
Průměrný rozdíl ve velikosti výsledku pro výběry řádků: 946						

Kolekce obsahuje milión řádků a rozsahový dotaz je sedm. Byl spuštěn výpočet odhadu v mé aplikaci a ukázal výsledek. Doplním ho do následující tabulky 9. Provedl jsem 7 dotazů. Průměr je pro výběry řádků dotazů.

Kolekce obsahující čísla s reálnými doménami se selektivitou má 10000 řádků a rozsahový dotaz je vybrán sedm z 1000 těchto dotazů. Byl proveden proces odhadu pro 1000 sekvence dotazu v mé aplikaci a měl jsem výsledek, který je v této tabulce 10. Průměrná chyba je pro všechny řádky dotazů.

7.4.5 Vizualizace

V aplikaci je k dispozici malá prezentace pro ukázku, jak vypadá pohyb bucket v průběhu výpočtu na horizontální a vertikální ose ve dvourozměrné poloze. Na ose je omezený rozsah v intervalu podle hodnoty pole X a Y, který závisí na vlastnosti ze vstupní datové sady. V této poloze jsou tečkovaně znázornění modré body, které obsadily pozici podle hodnoty X a Y. Na základě sekvenčního vstupního dotazu s požadovanými rozsahy z dvou dimenzionální polí se vykreslují objekty (bucket). V obdélníkovém objektu obsahují kolekce, které jsou umístěny se okolí hranice tohoto objektu. Z toho vychází počet frekvencí. Na vykreslení objektu jsou k dispozici různé barevné efekty v ohraničení čar. Na tomto obrázku můžeme vidět, že zelené ohraničení rámečku je mateřský bucket (nadřízený), modrou barvu má ten bucket, který je potomek (podřízený) rodičů. A dále zelená barva je od nadřízeného bucket, který má svou označenou modrou barvu.



Obrázek 14: Ukázka vykreslení bucket ve vizualizaci

Tabulka 10: STHoles: Srovnání mezi odhadem a skutečností pro desetitisící záznamy s reálnými doménami čísla.

Dimenze	2					
Kolekce	10000		**Dotaz**	7 z 1000 (Selektivita 0,2)		
Rozsah kolekce	0 .. 1		**Typ kolekce**	Číslo s reálnými doménami (Double)		
Výběr dotazu	**Rozsahový dotaz**	**Odhad**	**Skutečnost**	**Čas odhadu (ms)**	**Čas vykonání dotazu (ms)**	**Relativní chyba (%)**
7	[0,65;0,69] [0;0,53]	200	214	0	8	6,5
234	[0,68;0,95] [0,56;0,64]	195	195	0	11	0
312	[0,26;0,77] [0,03;0,07]	184	195	0	10	5,6
499	[0,07;0,20] [0,57;0,73]	194	190	0	9	2,1
555	[0,32;0,87] [0,01;0,04]	180	188	0	11	4,2
615	[0;0,04] [0,37;0,92]	189	207	0	7	8,7
799	[0,06;0,89] [0,30;0,32]	203	197	0	13	3
Průměrný rozdíl ve velikosti výsledku pro všechny řádky: 9						

8 Závěr

Jak již bylo napsáno v úvodu, zde je řešení rozsáhlé a komplikované problematiky s odhadem na vícerozměrném rozsahovém dotazu. Cílem této práce bylo získat potřebné odborné informace týkající se odhadu a nechat se inspirovat novými věcmi v teoretické části. Dále vytvořit vlastní implementaci s vlastní metodami odhadu a konečný výsledek na výstupu v praktické části. Probíral jsem všechny zdroje externí literatury a podle této literatury jsem aplikoval vlastní funkcionality pro řešení odhadu velikosti výsledku.

V teoretické části byly také popsány základní metody odhadu velikosti výsledku dotazu. Texty u všeobecně používaných metod jsou probrané zdroje z externí literatury. Konkrétně jde o dvě skupiny selektivity. Jedna skupina se nazývá jednorozměrný a druhá se jmenuje vícerozměrný histogram. U jednorozměrné kolekce jsou popsány zejména odborné stránky statistických vlastností. Jde o popisné charakteristiky a bodové odhady používané pro rozšíření výsledků nezávislého sběru dat na celou vyšetřovanou populaci. Požadovaná vybraná metoda na výsledky z odhadu byla samplování, které bylo aplikováno i v implementační praxi, protože výběr vzorků je běžně nejpoužívanější, proto jsou v mé práci tyto texty rozsáhlejší. Taktéž jsou popsány různé techniky odhadu. Ve vícerozměrné kategorii jsou popsány základní metody pro odhad na multidimenzionálních polích. Dokonce jsou uvedeny vzorce pro výpočty u těchto jednotlivých funkcí.

V praktické části byla aplikována implementace na metodu pro odhad velikosti výsledku ve vícerozměrném prostoru. Bylo naimplementováno ve vývojovém prostředí pro programovací jazyk C#. U implementace vytvoření vlastní aplikace je k dispozici několik ukázkových obrázků v textu a celý zdroj kódů je k dispozici v příloze. Obě metody jsou umístěny na jednotlivých záložkách. Metoda samplování funguje pro odhad pro jednorozměrný prostor. Zadané hodnoty ve formuláři na tuto záložku v aplikaci se naplňují podle potřeby (jestli známe rozptyl nebo neznáme), dále základní doplnění hodnoty směrodatné odchylky, počet vzorků, pole spolehlivosti se nastaví běžné nejméně, 95%, protože je to ve statistice nejpoužívanější. Dokonce jsou k dispozici ovládací prvky na vstupních souborech s daty a dotazy a výstup pro výsledek se provádí pomocí tlačítka odhadu v této aplikaci. Konečný výsledek, porovnání, průběh času v procesu pro výpočet bucket na odhad frekvence je k dispozici v informační zprávě aplikace. Pohyb barevných bucket se tu projevil vizuálně.

Literatura

[1] CHAUDHURI, Surajit; GRAVANO, Luis. Evaluating top-k selection queries. In: *In Proceedings of VLDB*. 1999. p. 397-410.

[2] BRUNO, Nicolas; CHAUDHURI, Surajit; GRAVANO, Luis. Top-k selection queries over relational databases: Mapping strategies and performance evaluation. *ACM Transactions on Database Systems (TODS)*, 2002, 27.2: 153-187.

[3] FUJIWARA, Yasuhiro, et al. Fast and exact top-k search for random walk with restart. *Proceedings of the VLDB Endowment*, 2012, 5.5: 442-453.

[4] POOSALA, Viswanath; IOANNIDIS, Yannis E. Selectivity estimation without the attribute value independence assumption. In: *In Proceedings of VLDB*. 1997. p. 486-495.

[5] POOSALA, Viswanath, et al. Improved histograms for selectivity estimation of range predicates. In: *ACM SIGMOD Record*. ACM, 1996. p. 294-305.

[6] Vapnik, Vladimir, Steven E. Golowich, and Alex Smola. "Support vector method for function approximation, regression estimation, and signal processing."Advances in neural information processing systems (1997): 281-287.

[7] CHRISTODOULAKIS, Stavros. Estimating record selectivities. *Information Systems*, 1983, 8.2: 105-115.

[8] CHEN, Chungmin Melvin; ROUSSOPOULOS, Nick. *Adaptive selectivity estimation using query feedback*. ACM, 1994.

[9] HAAS, Peter J., et al. Sampling-based estimation of the number of distinct values of an attribute. In: *In Proceedings of VLDB*. 1995. p. 311-322.

[10] CHAUDHURI, Surajit; GRAVANO, Luis. Optimizing queries over multimedia repositories. In: *ACM SIGMOD Record*. ACM, 1996. p. 91-102.

[11] FUCHS, Dennis; HE, Zhen; LEE, Byung Suk. Compressed histograms with arbitrary bucket layouts for selectivity estimation. *Information Sciences*, 2007, 177.3: 680-702.

[12] GUNOPULOS, Dimitrios, et al. Approximating multi-dimensional aggregate range queries over real attributes. In: *ACM SIGMOD Record*. ACM, 2000. p. 463-474.

[13] WHANG, Kyu-Young; KRISHNAMURTHY, Ravi. *The multilevel grid file: a dynamic hierarchical multidimensional file structure*. Korea Advanced Institute of Science and Technology, Center for Artificial Intelligence Research, 1991.

[14] MURALIKRISHNA, M.; DEWITT, David J. *Equi-Depth Multi-Dimensional Histograms*. University of Wisconsin-Madison, Computer Sciences Department, 1987.

[15] Golub, Gene H., and Christian Reinsch. "Singular value decomposition and least squares solutions."Numerische Mathematik 14.5 (1970): 403-420.

[16] KLEMA, Virginia; LAUB, Alan J. The singular value decomposition: Its computation and some applications. *Automatic Control, IEEE Transactions on, 1980*, 25.2: 164-176.

[17] ALBER, Jochen; NIEDERMEIER, Rolf. On multi-dimensional hilbert indexings. In: *Computing and Combinatorics*. Springer Berlin Heidelberg, 1998. p. 329-339.

[18] BRIŠ, Radim; LITSCHMANNOVÁ, Martina. *Statistika I.* Pro kombinované a distanční studium, VŠB-TU Ostrava, 2004.

[19] BRIŠ, Radim; LITSCHMANNOVÁ, Martina. *Statistika II.* Vysoká škola báňská-Technická univerzita, 2008.

[20] BRUNO, Nicolas; CHAUDHURI, Surajit; GRAVANO, Luis. STHoles: a multidimensional workload-aware histogram. In: *ACM SIGMOD Record*. ACM, 2001. p. 211-222.

[21] SELINGER, P. Griffiths, et al. Access path selection in a relational database management system. In: *Proceedings of the 1979 ACM SIGMOD international conference on Management of data*. ACM, 1979. p. 23-34.

[22] FUCHS, Dennis; HE, Zhen; LEE, Byung Suk. Compressed histograms with arbitrary bucket layouts for selectivity estimation. *Information Sciences*, 2007, 177.3: 680-702.

[23] NIEVERGELT, Jürg; HINTERBERGER, Hans; SEVCIK, Kenneth C. The grid file: An adaptable, symmetric multikey file structure. *ACM Transactions on Database Systems (TODS)*, 1984, 9.1: 38-71.

[24] PRESS, William H., et al. *Numerical recipes in C. Cambridge: Cambridge university press, 1996.*

[25] LEACH, Sonia. Singular value decomposition-a primer. 2006.

[26] MURALIKRISHNA, M.; DEWITT, David J. Equi-depth multidimensional histograms. In: ACM SIGMOD Record. ACM, 1988. p. 28-36.

[27] BÖHM, Christian, et al. Selectivity estimation of high dimensional window queries via clustering. In: *Advances in Spatial and Temporal Databases.* Springer Berlin Heidelberg, 2005. p. 1-18.

[28] ABOULNAGA, Ashraf; CHAUDHURI, Surajit. Self-tuning histograms: Building histograms without looking at data. *ACM SIGMOD Record*, 1999, 28.2: 181-192.

[29] Visual Studio - Microsoft Developer Tools <https://www.visualstudio.com>

[30] COHEN, Jacob. Statistical Power Analysis for the Behavioral Sciences. 2nd edn. Hillsdale, New Jersey: L. 1988.

[31] The New York Times <http://www.nytimes.com/1998/02/07/nyregion/jacob-cohen-74-psychologist-and-pioneer-in-statistical-studies.html>

[32] HEDGES, Larry V.; OLKIN, Ingram. *Statistical methods for meta-analysis.* Academic press, 2014.

[33] COOPER, Harris; HEDGES, Larry V.; VALENTINE, Jeffrey C. (ed.). *The handbook of research synthesis and meta-analysis.* Russell Sage Foundation, 2009.

A Přílohy diplomové práce

1. **Seznam vstupních i výstupních (report) souborů**, které jsou určeny na výstupy jako výsledek pro experimenty v mé aplikaci. Tento seznam je uložený v elektronické podobě na přiloženém DVD.

2. **Vizualizace bucket** metody STHoles, která je uložena jako obrázek v plném rozlišení na přiloženém DVD.

3. **DVD**, obsahující elektronické verze dokumentu této práce, proveditelný soubor a jeho knihovna `STholes.dll` pro zpracování odhadu vícerozměrného prostoru.